



**CENTRO UNIVERSITÁRIO DE BRASÍLIA -UniCEUB**  
**CURSO DE ENGENHARIA DE COMPUTAÇÃO**

**THIAGO ALENCAR FARIAS NEPOMUCENO**

**Sistema de Identificação de Veículos Automotores Utilizando Tecnologia  
RFID.**

**Orientador: Professora. M.C. Maria Marony Sousa Farias**

Brasília  
Junho, 2011

**THIAGO ALENCAR FARIAS NEPOMUCENO**

**Sistema de Identificação de Veículos Automotores Utilizando Tecnologia  
RFID.**

Trabalho apresentado ao Centro  
Universitário de Brasília  
(UniCEUB) como pré-requisito  
para a obtenção de Certificado de  
Conclusão de Curso de Engenharia  
de Computação.

Orientador: Professora M.C. Maria

Marony Sousa Farias

Este Trabalho foi julgado adequado para a obtenção do Título de Engenheiro de Computação,  
e aprovado em sua forma final pela Faculdade de Tecnologia e Ciências Sociais Aplicadas -  
FATECS.

---

Prof. Abiezer Amarilia Fernandez  
Coordenador do Curso

**Banca Examinadora:**

---

Professora M.C. Maria Marony Sousa Farias  
Orientador

---

Prof. Joao Marcos Souza Costa

---

Prof. Jose Julimá Bezerra Junior

---

Prof. Luciano Duque

**THIAGO ALENCAR FARIAS NEPOMUCENO**

**Sistema de Identificação de Veículos Automotores Utilizando Tecnologia  
RFID.**

Trabalho apresentado ao Centro  
Universitário de Brasília  
(UniCEUB) como pré-requisito  
para a obtenção de Certificado de  
Conclusão de Curso de Engenharia  
de Computação.

Orientador: Professora M.C. Maria  
Marony Sousa Farias

Brasília

Junho, 2011

**Dedico este trabalho ao meu pai, Vladimir Nepomuceno, e minha mãe, Ireny Alencar, pelo apoio incondicional. E a meu irmão, Vitor Nepomuceno que sempre esteve ao meu lado.**

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, por ter me dado à força de vontade necessária para concluir, por estar guiando minha vida e me confortando em todos os momentos. Agradeço, também, aos meus amigos Reinaldo Oliveira, Camila Oliveira e Marco Aurelio e todos os outros, que estiveram junto comigo durante esta etapa de minha vida. Aos professores Francisco Diaz, Maria Marony e Miguel Archanjo que me auxiliaram desde o início desta monografia. E por fim agradeço a minha família que proporcionou este momento e sempre me apoiaram nas minhas decisões.

## SUMÁRIO

<b>LISTA DE FIGURAS.....</b>	<b>20</b>
<b>LISTA DE QUADROS.....</b>	<b>22</b>
<b>LISTA DE ABREVIATURAS E SIGLAS .....</b>	<b>23</b>
<b>CAPÍTULO 1 - INTRODUÇÃO.....</b>	<b>24</b>
1.1– O Problema .....	13
1.2 – Objetivos do Trabalho .....	13
1.2.1- Objetivo Geral .....	13
1.2.2 – Objetivos específicos .....	14
1.3 – Justificativa e Importância do Trabalho .....	14
1.4 – Escopo do Trabalho .....	15
1.5 – Resultados Esperados .....	15
1.6 – Estrutura do Trabalho .....	16
<b>CAPÍTULO 2 – APRESENTAÇÃO DO PROBLEMA .....</b>	<b>17</b>
2.1 – O Problema .....	17
2.2 - Soluções Existentes .....	23
2.2.1 – Sistema Nacional de Identificação Automática de Veículos.....	23
2.2.2 – Sem parar/ via fácil.....	25
2.2.3 - Sinalerta.....	25
2.3 - Benefícios da Solução Apresentada .....	26
<b>CAPÍTULO 3 – BASES METODOLÓGICAS PARA RESOLUÇÃO DO PROBLEMA .....</b>	<b>27</b>
3.1 – Identificação por Rádio Frequência.....	27
3.1.1 – Introdução .....	27
3.1.2 – Histórico .....	28
3.1.3 - Funcionamento do RFID .....	29
3.1.4 – Componentes do RFID .....	31
3.1.6 – Por que utilizar o RFID?.....	38
3.2 – Linguagem de Programação – JAVA.....	39
3.2.1 – Introdução .....	39
3.2.2 – Conceitos sobre <i>Java</i> .....	39
3.3 – Banco de dados MySQL.....	43
3.3.1 – Banco de dados .....	43
3.3.2 – <i>MySQL</i> .....	43
<b>CAPÍTULO 4 – SISTEMA PROPOSTO.....</b>	<b>46</b>

4.1 – Apresentação Geral do Modelo Proposto .....	46
4.2 – Descrição das Etapas do Modelo .....	47
4.3 – Descrição da Implementação .....	48
4.3.1 – <i>Hardware</i> utilizado .....	48
4.3.2 – Metodologias e ferramentas utilizadas. ....	53
<b>CAPÍTULO 5 – APLICAÇÃO DO MODELO PROPOSTO .....</b>	<b>60</b>
5.1 - Apresentação da área de Aplicação do Modelo Proposto .....	60
5.1.1 - O ambiente de simulação .....	61
5.2 - Descrição da Aplicação do Modelo .....	62
5.2.1 - Módulo de cadastro .....	63
5.2.2 - Módulo de monitoramento .....	65
5.2.3 - Módulo de controle .....	66
5.3 - Resultados da aplicação do Modelo .....	67
5.4 - Custos do Projeto .....	69
5.5 - Avaliação Global do Modelo Proposto .....	70
<b>CAPÍTULO 6 – CONCLUSÃO .....</b>	<b>72</b>
6.1 - Conclusões .....	72
6.2 - Sugestões Para Trabalhos Futuros .....	73
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>74</b>

## LISTA DE FIGURAS

Figura 2.1 – Cadeia Produtiva (Plano de Logística CNT, 2010).....	19
Figura 2.3 – Elementos básicos do SINIAV (Seminário DENATRAN, 2010) .....	25
Figura 2.4 – Sem Parar/Via fácil (Sem Para/Via Fácil, 2010) .....	26
Figura 3.1 – Exemplo de aplicação utilizando RFID (Glover, 2006) .....	29
Figura 3.2 – Esquema básico de sistemas RFID (Congresso RFID Brasil, 2010) .....	31
Figura 3.3 – Sistema RFID integrado (Congresso RFID Brasil, 2010).....	31
Figura 3.4 – Tipos de etiquetas RFID (Phidgets, 2009) .....	32
Figura 3.5 – Componentes da Etiqueta (John Wiley & Sons, 2007).....	33
Figura 3.6 – Leitor RFID com Saída RS232 e ethernet (Shenzen Marktrace Co, 2010) .....	36
Figura 3.7 – Funcionamento de um leitor (Sanghera, 2007) .....	37
Figura 3.8 – Fases do programa em Java (Java como Programar, 2007) .....	42
Figura 4.1 – Etapas do Modelo.....	50
Figura 4.2 – Leitor <i>Phidget</i> RFID ( <i>Phidget</i> , 2011) .....	51
Figura 4.3 - Leitor RFID acoplado no computador .....	54
Figura 4.4 - Pannel de Controle Phidget .....	55
Figura 4.5 – Teste Leitor RFID .....	56
Figura 4.6 – Ferramentas Utilizadas.....	57
Figura 4.7 – Pacotes utilizados no desenvolvimento do software .....	58
Figura 4.8 – Pacote “SISMOVE.Forms” .....	58
Figura 4.9 – Pacote SISMOVE.Logica .....	59
Figura 4.10 – Pacotes “SISMOVE.Conexao” e “SISMOVE.SQL” .....	59
Figura 4.11 – Pacote MiddlewareRFID.....	59
Figura 4.12 – Pacotes “ListenersSC” e “ListenersSM” .....	60



Figura 4.13 – Banco de dados utilizado .....	60
Figura 5.1 – Ilustração da Maquete utilizada .....	65
Figura 5.2 – Cadastro de Veículos.....	67
Figura 5.3 – Módulo de Monitoramento .....	68
Figura. 5.4 – Módulo de Controle .....	69
Figura 5.5 – Interface do <i>Software</i> .....	70
Figura 5.6 – Maquete Utilizada .....	70
Figura 5.7 – Etiqueta RFID anexada ao veículo.....	71
Figura 5.8 – Conexão do leitor RFID com o computador .....	72

## **LISTA DE TABELAS**

Tabela 2.1 – Quantidade de Veículos por Estado – 2010.....	22
Tabela 3.1 – Faixas de Frequência RFID .....	38
Tabela 3.2 – Frequências, distância e tipo de etiqueta .....	39
Tabela 4.1 – Características do Leitor RFID.....	53

## **LISTA DE ABREVIATURAS E SIGLAS**

**EPC** – Electronic Product Code

**GPS** – Global Positioning System

**HF** – High Frequency

**LED** – Light Emitting Diode

**LF** – Low frequency

**RFID** – Radio Frequency Identification

**SGBD** – Sistema Gerenciador de Banco de Dados

**SPP** – Standard Parallel Port

**SQL** – Structured Query Language

**UHF** – Ultra high frequency

**USB** – Universal Serial Bus

**SNIAV** – Sistema Nacional de Identificação Automática de Veículos

**DENATRAN** – Departamento Nacional de Trânsito

**CONTRAN** – Conselho Nacional de Trânsito

## **RESUMO**

O projeto apresentado está dentro de uma grande área denominada infraestrutura de transportes. Neste projeto, por meio de um protótipo, é proposto o monitoramento de uma via pública de trânsito, utilizando-se da tecnologia RFID (identificação por rádio frequência), um sistema desenvolvido em linguagem de programação Java e com um banco de dados. Com a importância que área de transporte tem para o desenvolvimento do país e com o crescente número de veículos automotores circulando nas vias de trânsito do Brasil, é evidente a necessidade do desenvolvimento de um sistema computadorizado que consiga monitorar uma via de forma totalmente automática. Os objetivos alcançados demonstram que é possível atingir tal meta gerando maior segurança para proprietários de veículos e aumentando as opções para o monitoramento e controle do tráfego no país.

### **Palavras Chave:**

Identificação por rádio frequência, infraestrutura de transportes, segurança, trânsito.

## **ABSTRACT**

The project is presented within a large area called the transportation infrastructure. In this project, through a prototype, it is proposed the monitoring of a public road traffic, using RFID (radio frequency identification), a system developed in Java programming language and a database. With the importance of the transport area is to develop the country and the increasing number of motor vehicles circulating in the traffic lanes of Brazil, is a clear need to develop a computerized system that can monitor via a fully automatically. The results attained show that this goal can be achieved leading to greater safety for vehicle owners and increasing the options for monitoring and controlling the traffic in the country.

### **Keywords:**

Radio frequency identification, transport infrastructure, security, traffic.

## **CAPÍTULO 1 - INTRODUÇÃO**

### **1.1– O Problema**

Os sucessivos recordes de trânsito no dia-a-dia do brasileiro que mora em grandes cidades, como Brasília e São Paulo, demonstram que a malha viária dessas cidades já não suporta o crescimento acentuado da frota de veículos. Dados estatísticos do Departamento Nacional de Trânsito – DENATRAN mostra que somente no estado de São Paulo são mais de vinte milhões de veículos registrados. No Distrito Federal esse número chega a pouco mais de um milhão de veículos registrados. Decorrente deste aumento desenfreado da frota de veículos nacional surge um problema que afeta todas as grandes cidades: a falência da infraestrutura de transporte e logística no país.

É preciso pensar urgentemente em um projeto para que o trânsito no Brasil ganhe qualidade. Sem a logística adequada, o desafio de o Brasil ser o protagonista na nova ordem econômica mundial ficará mais complicado e difícil de ser alcançado, devido à dificuldade encontrada no escoamento de suas mercadorias.

Indo na direção da qualidade da infraestrutura do transporte, o trabalho apresenta uma solução para o monitoramento e controle do trânsito em todas as cidades brasileiras. Hoje é impossível implementar um sistema de monitoramento e controle do trânsito sem que haja algum tipo de tecnologia envolvida. Dessa forma o trabalho tenta responder a seguinte questão: como monitorar e controlar uma via de trânsito, por onde passam milhares de veículos todos os dias?

### **1.2 – Objetivos do Trabalho**

#### **1.2.1- Objetivo Geral**

Este trabalho visa propor uma solução, por meio de um protótipo, que atue no monitoramento e, conseqüentemente, apoie o gerenciamento, planejamento e fiscalização de

um dos piores problemas de todas as grandes cidades do Brasil: o trânsito. Visando auxiliar e ser mais uma ferramenta de controle, disponível para aumentar a segurança no trânsito o trabalho explora uma fragilidade ainda muito grande na malha viária do Brasil. Utilizando a tecnologia *RFID* (*Radio Frequency Identification*), será apresentada uma simulação de uma via monitorada através de um sistema desenvolvido para esta finalidade.

### 1.2.2 – Objetivos específicos

Para alcançar o objetivo geral do projeto foram traçados alguns objetivos específicos que primeiramente precisam ser alcançados:

- Identificar de forma única cada veículo que trafega na via;
- Desenvolver um sistema capaz de monitorar o tráfego na via;
- Integrar o sistema com a tecnologia RFID;
- Disponibilizar corretamente os dados armazenados no banco de dados, quando o veículo for identificado; e
- Armazenar a informação dos veículos que passaram na via.

## 1.3 – Justificativa e Importância do Trabalho

Com base na resolução nº 212 de 2006 do CONTRAN (Conselho Nacional de Trânsito), que trata da implantação do SINIAV (Sistema Nacional de Identificação Automática de Veículos) – ficou claro a necessidade de um sistema computadorizado que conseguisse monitorar uma via de trânsito pública de forma totalmente automática. O protótipo apresentado tende a gerar mais segurança ao proprietário do veículo, uma vez que, verifica os dados do veículo sem a necessidade de pará-lo. Assim aumentam-se as chances de localizar e reter veículos em situações irregulares, ou até mesmo, no caso de roubo e furto.

## **1.4 – Escopo do Trabalho**

O projeto abrange todo o processo de identificação de veículos em circulação. Busca por meio da tecnologia RFID, reconhecer 100% dos veículos que transitaram em determinado ponto da via monitorada. Não se busca aqui desenvolver mais um dispositivo de fiscalização do trânsito, como pardais e barreiras eletrônicas, mas sim um equipamento com o intuito de auxiliar na fiscalização e na gestão do trânsito. Está dentro do escopo do projeto identificar de forma única todos os veículos presentes na simulação, como também armazenar o histórico do ponto de monitoramento e controle que foi previamente acertado.

O trabalho não trata de solucionar alguns problemas encontrados durante sua execução, tais como: determinação da posição exata do veículo num dado instante; aplicação de multas de trânsito, identificação do condutor do veículo; ser utilizado como GPS (Sistema de Posicionamento Global), identificação de possíveis congestionamento na via e procurar rotas alternativas. Com isso o escopo do trabalho é identificar qual veículo está trafegando na via e armazenar o histórico do tráfego em determinado ponto da via.

## **1.5 – Resultados Esperados**

O projeto final tem como principais resultados o monitoramento e controle do tráfego, a atualização do sistema, e disponibilização do histórico da via. Dessa forma, o protótipo final desenvolvido visa atuar como uma ferramenta auxiliar para o monitoramento do tráfego na via. Abaixo são destacados alguns resultados esperados:

- Possibilidade de atualização do sistema, como por exemplo, mudança de proprietário;
- Cadastrar, caso necessário, veículos novos;
- Disponibilizar data e hora que o veículo passou pela barreira de identificação; e
- Possibilitar a exclusão do veículo no caso de baixa do veículo.



## 1.6 – Estrutura do Trabalho

Além deste capítulo introdutório, a monografia está estruturada em mais cinco capítulos e organizada da seguinte forma:

- Capítulo 2 – Apresentação do Problema: Nesse capítulo é apresentada uma descrição do problema que o projeto pretende resolver. Primeiramente serão demonstrados alguns dados atuais e como o problema vem sendo resolvido por alguns profissionais da área. Em seguida são mostrados os benefícios da solução apresentada.
- Capítulo 3 – Bases Metodológicas para a Resolução do Problema: Aqui a monografia apresenta todo o referencial teórico e tecnológico que embasa o projeto. Além de aplicar conteúdos vistos nas disciplinas para a resolução do problema proposto.
- Capítulo 4 – Modelo Proposto: Esse capítulo detalha todas as etapas e passos necessários para a resolução do problema apresentado no capítulo 2.
- Capítulo 5 – Aplicação do Modelo Proposto – É apresentado nesse capítulo uma aplicação prática, envolvendo um caso real de aplicação, mostrando a viabilidade da proposta de resolução sugerida. Aqui são demonstrados os testes, a simulação e a avaliação do projeto.
- Capítulo 6 – Conclusão – Aqui é o final da monografia, permitindo uma análise completa de todo o desenvolvimento do projeto e da monografia. Em seguida são apresentadas sugestões para trabalhos futuros.

## **CAPÍTULO 2 – APRESENTAÇÃO DO PROBLEMA**

### **2.1 – O Problema**

O Brasil vive um momento único, jamais visto em sua história. Depois de anos e anos de imobilismo, o País acordou para a necessidade de se investir em infraestrutura de transportes como fator essencial para estimular e garantir o crescimento econômico. E mais, o País percebeu que o transporte produz riqueza, gera empregos e fortalece a indústria, a pecuária, a agricultura e os serviços. Diante do seu papel de indutor do crescimento, o Governo adotou medidas que aumentaram significativamente os recursos para o setor, como o Programa de Aceleração do Crescimento – PAC (Plano de Logística da Confederação Nacional dos Transportes, 2010).

A importância do setor de transportes ultrapassa o simples conceito de meio de deslocamento de bens e pessoas. É também um agente indutor de riqueza e desenvolvimento, uma vez que gera emprego, transferência econômica e consome produtos e insumos de outros setores – como combustíveis, veículos e peças, entre outros. Entretanto, a infraestrutura de transportes brasileira tem sido considerada como um grande entrave ao desenvolvimento econômico do País.

A demanda por bens movimenta a cadeia produtiva, que vai desde a obtenção de matéria-prima, passa pela produção e comercialização e finaliza com o consumo – que é exatamente dependente dos custos envolvidos nos processos de produção e na movimentação dos itens que constituem esses bens em todas as fases do processo produtivo, como mostra a figura a seguir (Plano de Logística da Confederação Nacional dos Transportes, 2010).

Sabe-se que o custo de transporte representa a maior parte do custo logístico total das empresas. Com isso além dos fatores referentes à eficiência da gestão, o desempenho das operações logísticas está relacionado à qualidade, à oferta e à segurança da infraestrutura viária, de veículos e terminais.



Figura 2.1 – Cadeia Produtiva (Plano de Logística da Confederação Nacional dos Transportes, 2010)

A infraestrutura viária, em seu nível mais amplo, compreende o tráfego de veículos de carga e o conjunto de vias – rodovias, ferrovias, hidrovias e aerovias. Geralmente essa infraestrutura é planejada e financiada pelos Governos Estaduais e Federal. Já em um nível urbano, mais reduzido, essa infraestrutura abrange o tráfego de veículos de carga e de passageiros e o conjunto de vias locais – rodovias (ou vias urbanas), ferrovias (metrô e trens urbanos) e aquavias (sistemas de barca), sendo projetada e financiada tanto pelos Governos Municipais, como pelos Governos Estaduais e Federal.

Na ocorrência de um sistema de transporte ineficiente, observa-se que:

- Nos setores econômicos produtivos, as ineficiências agregam custos, aumentando o preço final do produto a ser comercializado;
- A população economicamente ativa, responsável pelo consumo de produtos, arca com essa ineficiência logística em razão dos custos embutidos nos valores dos bens consumidos;

- Para o meio ambiente, os custos da ineficiência da malha de transporte acarretam altos índices de emissão de poluentes;
- A sociedade brasileira perde como um todo, pois a ineficiência acarreta uma baixa competitividade dos produtos brasileiros no mercado externo e, conseqüentemente, há uma menor geração de divisas. Além disso, observam-se as dificuldades na integração física entre cidades de diferentes Estados e até mesmo com os países vizinhos, bem como a queda no nível do serviço oferecido à população em geral, que precisa utilizar serviços de transporte.

No transporte rodoviário, verifica-se grande incidência de rodovias em precárias condições de conservação e funcionalidade, o que eleva os riscos de quebra mecânica dos veículos e de graves acidentes com grande número de vítimas. Ademais, a idade média da frota rodoviária é muito elevada – fato que contribui para uma grande emissão de poluentes, além da segurança em rodovias que atualmente é precária.

Além dessas questões de âmbito nacional, há, também, o tema metropolitano, que reflete a falta de políticas direcionadas para o transporte urbano e o incentivo cada vez maior do uso do transporte individual. Com isso, observa-se um crescimento significativo do número de veículos nos grandes centros urbanos, resultando na saturação das vias e gerando grandes congestionamentos – aumentando, dessa forma, os tempos e os custos das viagens.

A malha rodoviária brasileira tem atualmente uma extensão de 1.580.809 km, com apenas 212.618 km de pistas pavimentadas – o que representa aproximadamente de 13,4% da extensão total. Essas, por sua vez, estão distribuídas conforme a jurisdição da seguinte forma: 61.961 km de rodovias federais, 123.830 km de rodovias estaduais e 26.827 km de rodovias municipais (Departamento Nacional de Trânsito – Denatran, 2010).

A responsabilidade pela ampliação, conservação e manutenção da malha compete aos Governos Federal, Estaduais e Municipais, conforme a respectiva jurisdição. Contudo, esses podem, por meio de licitação, conceder trechos à iniciativa privada – seja para todos os serviços, seja apenas para a manutenção. Assim, hoje, em torno de 15.816 km das rodovias pavimentadas são administrados por operadoras estaduais e pela iniciativa privada, mediante a cobrança de tarifas de pedágio – revertidas para serviços de atendimento ao usuário, ampliação da capacidade e manutenção da malha rodoviária.

Com relação à frota de veículos rodoviários dados divulgados pelo Departamento Nacional de Trânsito – Denatran revelam que a frota de veículos brasileira teve aumento de 8,4 % em 2010, totalizando 64.817.974 veículos em todo o país. O gráfico abaixo apresenta o aumento de veículos em circulação de 2006 até 2010.

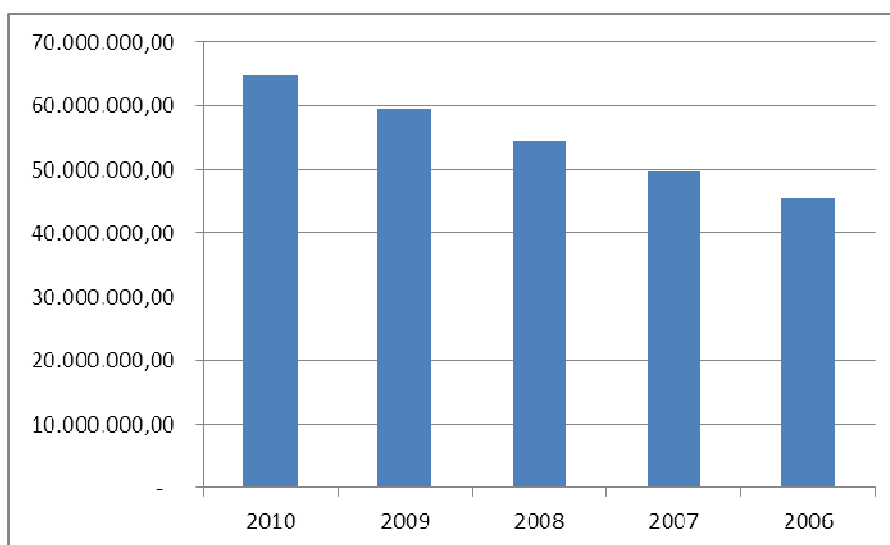


Gráfico 2.1 – Quantidade de Veículos Por Ano no País.

Fonte dos dados: DENATRAN

O estado de São Paulo lidera na quantidade de veículos (20.537.980), seguido por Minas Gerais (7.005.640), Paraná (5.160.354), Rio Grande do Sul (4.808.503) e Rio de Janeiro (4. 489.680). O Distrito Federal aparece na 12ª posição com 1.245.521 veículos registrados, como mostra a tabela 2.1.

Tabela 2.1 – Quantidade de Veículos por Estado – 2010

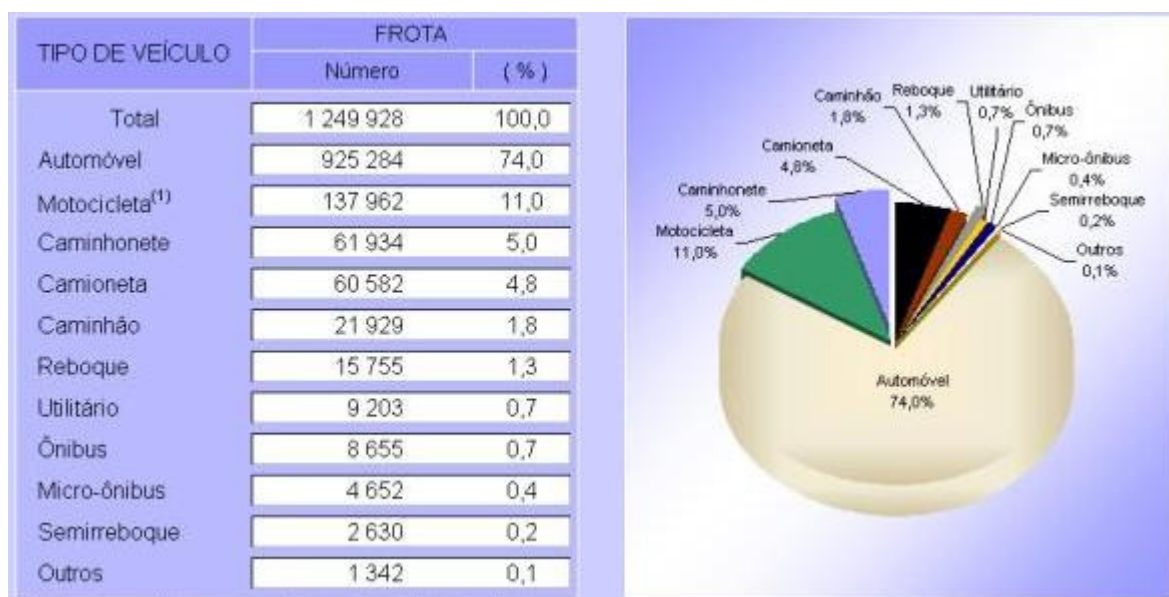
<b>Quantidade de Veículos por Estado - 2010</b>					
1º	São Paulo	20.537.980,00	15º	Pará	969.667,00
2º	Minas Gerais	7.005.640,00	16º	Maranhão	796.083,00
3º	Paraná	5.160.354,00	17º	Rio Grande do Norte	731.263,00
4º	Rio Grande do Sul	4.808.503,00	18º	Paraíba	698.556,00
5º	Rio de Janeiro	4.489.680,00	19º	Piauí	582.777,00
6º	Santa Catarina	3.414.195,00	20º	Rondônia	561.811,00
7º	Goiás	2.428.705,00	21º	Amazonas	530.814,00
8º	Bahia	2.308.978,00	22º	Alagoas	438.788,00
9º	Pernambuco	1.774.389,00	23º	Sergipe	427.048,00
10º	Ceará	1.711.998,00	24º	Tocantins	394.628,00
11º	Espírito Santo	1.262.848,00	25º	Acre	151.320,00
12º	Distrito Federal	1.245.521,00	26º	Roraima	125.451,00
13º	Mato Grosso	1.173.125,00	27º	Amapá	115.323,00
14º	Mato Grosso do Sul	972.529,00			
<b>TOTAL</b>					<b>64.817.974,00</b>

Fonte: Departamento Nacional de Trânsito – Denatran, 2011

Até dezembro de 2010, dados do Denatran mostram que eram 37.188.341 automóveis circulando nas vias do nosso país, correspondendo a 57,37% da frota total. Já as motocicletas somam 13.950.448 e são 21,52% da frota nacional. Na região Norte a frota de motocicletas ultrapassa a de automóveis em cinco estados: Acre, Pará, Rondônia, Roraima e Tocantins. No nordeste, Ceará, Maranhão e Piauí também possuem mais motocicletas que automóveis.

No Distrito Federal levantamento feito pelo Departamento de Trânsito do Distrito Federal - DETRAN-DF - mostra que existem cerca 1.249.928 veículos em circulação, até fevereiro de 2011. Sendo mais de 900 mil do tipo automóvel. São cerca de 2 pessoas para cada veículo registrado no DF (tomando por base a estimativa da população do DF, realizada em 2009 pelo IBGE, de 2.606.885 habitantes).

Quadro 2.1 – Frota do Distrito Federal



Fonte: DETRAN – DF, 2011

A tendência natural é que com o passar do tempo estes números aumentem cada vez mais, tornando o monitoramento e a identificação dos veículos algo complexo e trabalhoso.

Com o intuito de auxiliar no monitoramento e na identificação dos veículos que trafegam nas vias públicas, este projeto propõe, por intermédio de um protótipo, a construção de um sistema capaz de identificar veículos trafegando em um determinado ponto da via, de forma automatizada, utilizando a identificação por rádio frequência (RFID) e um sistema desenvolvido para informar as características do veículo. Assim será possível verificar os seus dados sem que seja necessário interromper o fluxo contínuo da via, ou para o veículo. A ideia principal do projeto é identificar de forma eletrônica os veículos em circulação.

Cada veículo terá uma etiqueta eletrônica que, quando solicitado, enviará o sinal a uma antena leitora que por sua vez irá enviar os dados coletados até um banco de dados, que analisa e, por exemplo, pode verificar se existe algum tipo de restrição contra aquele veículo. Dessa forma o sistema desenvolvido visa auxiliar na fiscalização urbana e na gestão do trânsito.

## 2.2 - Soluções Existentes

Atualmente os veículos são identificados por caracteres gravados no chassi e externamente por placa de identificação dianteira e traseira, sendo esta lacrada. Hoje todos os veículos após serem registrados recebem o Certificado de Registro do Veículo, documento este de porte obrigatório.

Com o intuito de modernizar os equipamentos e procedimentos empregados nas atividades de prevenção, fiscalização e repressão ao furto e roubo de veículos e considerando a necessidade de dotar os órgãos executivos de trânsito de instrumentos modernos e interoperáveis o CONTRAN (Conselho Nacional de Trânsito) instituiu o Sistema Nacional de Identificação Automática de Veículos – SINIAV, baseado em tecnologia de identificação por radiofrequência (RFID), composto por: placas de identificação veicular eletrônica, instaladas nos veículos, antenas leitoras, centrais de processamento e sistemas informatizados.

### 2.2.1 – Sistema Nacional de Identificação Automática de Veículos

A ideia inicial da proposta, de acordo com a resolução nº 212/2006 do CONTRAN, era que todos os veículos fossem identificados com etiquetas eletrônicas até maio de 2008, porém a resolução nº 338/2010 alterou o cronograma de implantação do SINIAV, hoje todos os Estados e o Distrito Federal têm até 30 de junho de 2011 para iniciar a implantação do SINIAV e o processo deverá ser concluído até o dia 30 de junho de 2014.

O SINIAV – Sistema Nacional de Identificação Automática de Veículos – foi criado devido à necessidade de modernizar os equipamentos e os procedimentos empregados nas atividades de prevenção, fiscalização e repressão ao furto de veículos e de carga, além da importância de dotar os órgãos executivos de trânsito de instrumentos modernos e interoperáveis para o planejamento, a fiscalização e gestão do trânsito e da frota de veículos.

O SINIAV é um sistema baseado em tecnologia de identificação por rádio frequência (RFID) e composto por:



- Placas de identificação veicular eletrônica - instaladas nos veículos;
- Antenas leitoras - que recebem e transmitem dados às placas de identificação veicular eletrônica, instaladas nos veículos no momento da passagem dos mesmos pela área de abrangência;
- Sistemas de transmissão; e
- Sistemas de processamento de dados.



Figura 2.3 – Elementos básicos do SINIAV (Seminário DENATRAN, 2010)

O leitor deverá desempenhar a leitura de pelo menos 99,9% dos veículos que trafegam na via e estão equipados com as placas de identificação, permitir a identificação inequívoca da faixa de rolamento em que se encontra o veículo, além de realizar a leitura das placas de identificação veicular instaladas nos veículos que estejam em qualquer velocidade dentro do intervalo de 0 até 160 km/h. (DETRAN DF, 2010)

Quanto ao processo de implantação do SINIAV a instalação da placa de identificação veicular eletrônica terá de ser iniciada em todos os Estados até o dia 30 de junho de 2011 e, como já foi dito, finalizada até o dia 30 de junho de 2014 e se dará:

- Para veículos novos no primeiro licenciamento, quando serão inseridos os dados permanentes;
- Para veículos usados e outros, no licenciamento anual.

### 2.2.2 – Sem parar/ via fácil

Hoje também existe o Sem Parar/Via Fácil. Um sistema que permite que o condutor não tenha de parar o veículo ao passar por uma praça de pedágio, usando um pequeno transmissor de rádio frequência colado ao para-brisa. Este sistema foi integrado para os Estados do Paraná, Minas Gerais, Rio Grande do Sul, Santa Catarina, Bahia, São Paulo e Rio de Janeiro. O sistema permite ao usuário utilizar estacionamentos de shoppings e aeroportos sem necessidade de utilização de tíquetes de estacionamento. Ao final do mês na data marcada o sistema do Sem Parar/Via Fácil emite uma fatura para pagamento em qualquer banco.



Figura 2.4 – Sem Parar/Via fácil

### 2.2.3 - Sinalerta

A empresa RaFID também utiliza da tecnologia RFID no desenvolvimento de seus produtos. O SINALERTA é um bom exemplo. Um equipamento desenvolvido para ser instalado no interior dos veículos (para-brisas), e que recebe sinais de perigo ou de acidente e alertam os motoristas por meio de mensagens visuais em um "display" e também de mensagens sonoras no alto-falante (voz digitalizada). Os sinais de perigo ou de acidente são enviados por antenas (fixas ou móveis) dispostas nos pontos considerados mais críticos pelas empresas que administram as rodovias.

Ou seja, uma vez gerado um sinal de perigo ou de acidente, de qualquer proveniência, cria-se um estado de atenção em torno desta região crítica e o motorista que recebe os alertas reduz aos poucos sua velocidade e passa a dirigir com maior segurança.

### **2.3 - Benefícios da Solução Apresentada**

A tecnologia de identificar o veículo e poder verificar sua documentação por leitura eletrônica sem a necessidade de para-lo traz a possibilidade de gerar maior segurança ao proprietário, facilidade no monitoramento da via e auxílio na fiscalização urbana, além de outros fatores.

Um exemplo prático seria o alerta à agentes de trânsito sobre veículos irregulares, possibilitando a substituição de bloqueios policiais de fiscalização, deixando de lado a seleção aleatória de veículos.

No que diz respeito ao proprietário podemos supor uma eventual diminuição no valor do prêmio para seguros veiculares, já que instalando leitores em diversos pontos da malha viária urbana e monitorando o tráfego constantemente haveria uma maior possibilidade de identificação de veículos roubados.

## **CAPÍTULO 3 – BASES METODOLÓGICAS PARA RESOLUÇÃO DO PROBLEMA**

Neste capítulo é apresentado todo o referencial teórico e tecnológico que é utilizado como base no projeto. Serão demonstrados todos os softwares, componentes de hardware, metodologias e as ferramentas utilizadas, além de justificar a escolha de cada item.

### **3.1 – Identificação por Rádio Frequência**

#### **3.1.1 – Introdução**

RFID significa *Radio Frequency Identification*, ou na sua tradução direta Identificação por Rádio Frequência, é um termo que descreve qualquer sistema de identificação no qual um dispositivo eletrônico que usa frequência de rádio para se comunicar é anexado a um item. A identificação por rádio frequência é uma tecnologia de comunicação que utiliza ondas de rádio, produzidas e enviadas por equipamentos eletrônicos, para a identificação de produtos, documentos, objetos, animais e até mesmo seres humanos. Por intermédio dela é possível rastrear e monitorar sem a necessidade de contato físico ou visual (Glover & Bhatt, 2007).

Os dois componentes mais importantes de um sistema RFID são o identificador, ou etiqueta eletrônica, que é anexado ao item que deve ser rastreado e o leitor, que é um dispositivo que consegue reconhecer a presença de etiquetas RFID e ler as informações armazenadas nelas.

A tecnologia RFID permite atribuir uma identidade única a um objeto através das etiquetas eletrônicas. Assim, quando um objeto recebe sua identificação eletrônica e entra na área de cobertura do leitor, este envia um sinal para que a etiqueta transmita os dados armazenados. As etiquetas podem armazenar e enviar vários tipos de informação sobre o objeto que estão identificando, incluindo números de série, instruções de configuração e características específicas do objeto (Glover & Bhatt, 2007).

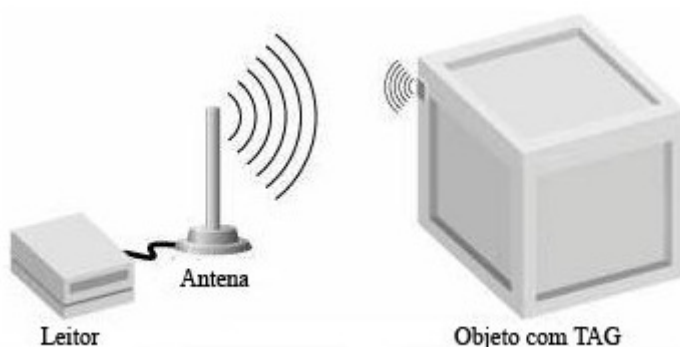


Figura 3.1 – Exemplo de aplicação utilizando RFID (Glover, 2007)

Atualmente apesar de poucas pessoas conhecerem a tecnologia ela se encontra presente em várias aplicações utilizadas por nós. Desde aplicações comerciais como, por exemplo, no controle de estoque e na segurança de mercadorias, até aplicações hospitalares no controle de funcionários, remédios e equipamentos.

### 3.1.2 – Histórico

O eletromagnetismo e o entendimento dos mecanismos das ondas de rádio foi uma das grandes descobertas do século XIX. Atribui-se aos chineses as primeiras observações de fenômenos magnéticos, utilizando ímãs naturais, cerca de 100 anos antes de Cristo. A partir das observações do físico *James Clerk Maxwell* que os avanços começaram a ocorrer rapidamente. No século XIX que foram feitas as primeiras transmissões telegráficas intercontinentais utilizando ondas de rádio, porém a onda não portava informação, sua presença ou ausência era codificada para ter significado (Universidade Federal do Rio de Janeiro – UFRJ, 2009).

Em 1906 que surgiram as primeiras ondas de rádio contínuas pelas mãos de *Ernst F. W. Alexanderson*, que permitiu controlar e empregar todas as características da onda para a transmissão da informação, possibilitando a transmissão em maior volume e com menor ocorrência de erros. O passo seguinte foi o surgimento do radar que emite ondas de rádio em todas as direções e a sua reflexão, quando captada, permite identificar a presença de objetos. Logo a tecnologia RFID tem suas raízes nos sistemas de radares amplamente utilizados na

Segunda Guerra Mundial pelos alemães, japoneses, americanos e ingleses (Universidade Federal do Rio de Janeiro – UFRJ, 2009).

O RFID surgiu inicialmente como solução para sistemas de rastreamento e controle de acesso na década de 80 quando o *MIT (Massachusetts Institute of Technology)*, juntamente com outros centros de pesquisa, iniciou o estudo de uma arquitetura que utilizasse os recursos das tecnologias baseadas em radiofrequência para ser utilizado no desenvolvimento de aplicações voltadas para o rastreamento e localização de produtos. Desse estudo nasceu o Código Eletrônico de Produtos – *EPC*. Foi o *EPC* que definiu a arquitetura de produtos que utilizavam os recursos proporcionados pelos sinais de radiofrequência, depois chamado de RFID. (Santana, 2009).

O RFID, atualmente, é amplamente utilizado pelas empresas e pessoas de todo o mundo para ajudar a reduzir o desperdício, limitar roubos, gerir inventários, simplificar a logística e até aumentar a produtividade. Exemplo de soluções que empregam a tecnologia RFID é a empresa Mitsubishi que utiliza no controle de produção gerando relatórios para a área gerencial e para o controle de estoque

### 3.1.3 - Funcionamento do RFID

O princípio de funcionamento da tecnologia RFID é relativamente simples como mostrado na figura 3.2. Sistemas RFID são compostos pela etiqueta eletrônica, leitor e um sistema com o qual o leitor se comunica, geralmente executando um software que fica entre o leitor e as aplicações, chamado de middleware.



Figura 3.2 – Esquema básico de sistemas RFID (Congresso RFID Brasil, 2009).

O leitor transmite sinais de rádio em uma frequência e intervalo pré-estabelecido, por intermédio da antena. Qualquer etiqueta eletrônica que estiver na faixa de frequência de rádio do leitor captará sua transmissão através de uma antena embutida que é capaz de escutar sinais de rádio na frequência pré-estabelecida. Por sua vez, as etiquetas eletrônicas, enviam de volta as informações armazenadas e programadas pra transmitir (Congresso RFID Brasil, 2009).

Leitores RFID operam pela emissão de um campo eletromagnético e são utilizados para reconhecer a presença de etiquetas RFID. Uma antena em um identificador próximo capta esta energia e o identificador então a converte em energia elétrica através de indução. Esta energia é suficiente para energizar o chip semiconductor, anexado à antena do identificador, que armazena os dados do identificador. O identificador então envia a identidade de volta para o leitor (John Wiley & Sons, 2007).

Esses elementos (leitor e identificador) são integrados a uma infraestrutura, que dá suporte a comunicação de sistemas de processamento que são responsáveis por manipular os dados lidos pelos leitores e os transformar em informação útil para o usuário final, os chamados middlewares. A figura 3.3 exemplifica esse processo.



Figura 3.3 – Sistema RFID integrado (Congresso RFID Brasil, 2009).

### 3.1.4 – Componentes do RFID

#### 3.1.4.1 – Etiquetas RFID

Como elemento principal na tecnologia RFID, a etiqueta eletrônica é o alvo preferencial das pesquisas e desenvolvimento. A arquitetura da etiqueta é direcionada pela aplicação que se destina, fazendo com que exista um modelo mais indicado para cada tipo de problema que se pretende solucionar.



Figura 3.4 – Tipos de etiquetas RFID (Phidgets, 2009)

Cada etiqueta possui algum mecanismo interno para armazenar dados e uma forma de comunicar esses dados, mas nem todas as etiquetas possuem um microchip ou uma fonte interna de energia. As principais características de uma etiqueta RFID são a presença de chip, a forma de alimentação, as características da memória e a frequência de operação. As mais sofisticadas costumam ter circuitos integrados, que gerenciam as principais funções no dispositivo, neste caso, são responsáveis por armazenar os dados na memória, implementar propriedades anticóllisão e oferecer funções de alto nível (registro de eventos, data e hora, por



exemplo). Enquanto são privadas de oferecer tais recursos, as etiquetas sem chip são consideravelmente mais baratas. Sua operação é mais simples e se baseia na simples reflexão da onda incidente, isto é, na manipulação das propriedades físicas do material para criar um padrão de resposta único. (Glover e Bhatt, 2007).

As etiquetas mais básicas consistem em chips eletrônicos, uma antena e o encapsulamento (pode ser usado plástico, vidro e epóxi), como é apresentado na figura abaixo. Da junção dos três que é formado a etiqueta e disponibilizado no mercado em diversos formatos e tamanhos.

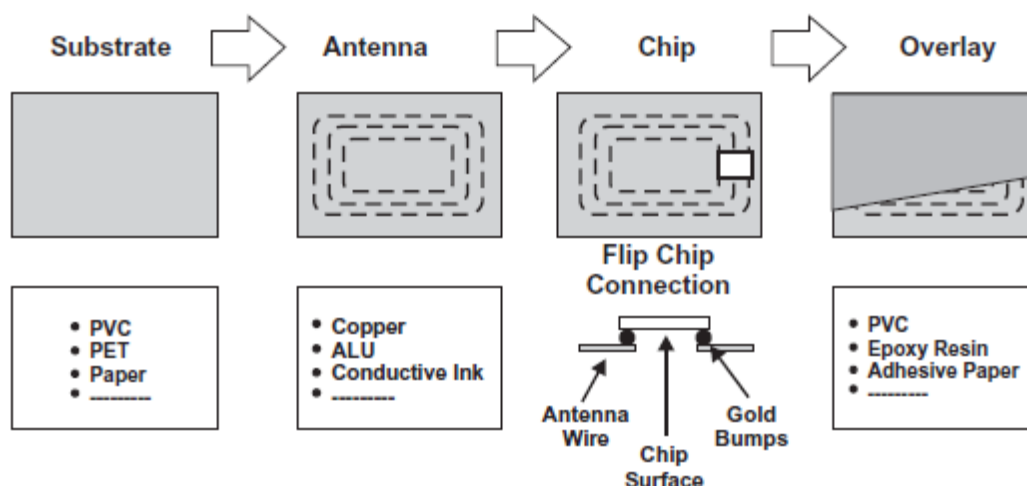


Figura 3.5 – Componentes da Etiqueta (John Wiley & Sons, 2007).

Várias etiquetas oferecem um ou mais dos seguintes recursos e capacidades apresentadas abaixo: (Glover e Bhatt, 2007)

1. Matar/desabilitar: Algumas permitem que um leitor as comande para pararem de funcionar permanentemente, ou seja, receber o “código de morte”.
2. Gravar uma vez: Etiquetas podem ser fabricadas com seus dados já gravados permanentemente na fábrica, como também um usuário final pode configurar um determinado valor para gravar uma única vez na etiqueta.
3. Regravação: Podemos regravar dados várias vezes em uma etiqueta, ou seja, a etiqueta permite apagar e escrever informações.

4. Anticolisão: Com esta capacidade sabem aguardar sua vez de se comunicar com o leitor, isso ocorre quando várias etiquetas estão próximas.
5. Segurança e criptografia: Etiquetas com essas características podem participar de comunicações criptografadas e algumas responderão apenas a leitores que puderem fornecer uma senha secreta.
6. Compatibilidade com padrões: Uma etiqueta pode ser compatível com um ou mais padrões, permitindo que se comunique com diversos leitores.

Um tipo muito comum de acoplamento entre o leitor e a etiqueta RFID é o acoplamento indutivo. Neste tipo de acoplamento um leitor fornece energia para etiquetas, utilizando uma antena espiral para gerar um campo magnético. O campo dirige a corrente por uma espiral no identificador por indução de forma semelhante a um transformador transfere energia entre duas bobinas.

O campo magnético gerado pelo leitor fornece energia suficiente para o microchip, o qual pode então se comunicar com o leitor pela modulação de carga. Um resistor que seja ligado e desligado no identificador gera flutuações no campo magnético, o que cria alterações de voltagem na antena do leitor.

Um modo muito comum de se categorizar etiquetas ou identificadores RFID é pela sua fonte de energia. Este também é um dos principais fatores determinantes para o custo e a longevidade do dispositivo. Atualmente podemos encontrar três categorias de identificadores, quanto sua fonte de energia:

- Etiquetas ativas – são alimentadas por uma bateria interna.
- Etiquetas passivas – operam sem bateria e são alimentadas pelo próprio leitor.
- Etiquetas semi-passivas – usam baterias para algumas funções, mas permitem que o leitor forneça energia para a transmissão.

As etiquetas RFID ativas utilizam a própria fonte de energia que possuem quando necessitam transmitir dados ao leitor, fazendo com que possa comunicar-se com leitores mais fracos e transmitir a informação armazenada a distâncias maiores. Além disso, esses tipos de

etiquetas RFID, tipicamente, podem armazenar maior quantidade de informação, podendo ter até 128 Bytes de memória. Contudo elas são maiores, mais complexas, e mais caras do que as etiquetas passivas (Finkenzeller, 2009).

Tais etiquetas permitem ainda processos de leitura e escrita, podendo iniciar a comunicação mandando seu próprio sinal não dependendo do leitor para utilizar seu circuito interno e para criar sinais. No que diz respeito a sua operação, elas podem tanto ficar ligada o tempo todo quanto serem “acordadas” assim que receber um sinal. Devido a estas características podem ser integradas a sistemas de posicionamento global (*Global Positioning System – GPS*) para a determinação de sua posição (Finkenzeller, 2009).

Ao contrário das etiquetas ativas as passivas por não possuírem uma fonte de alimentação interna, retiram a energia necessária para transmitir os dados, do sinal enviado do leitor. O resultado é um dispositivo menor, mais barato e menos complexo, no entanto com alcance muito menor do que as etiquetas ativas e possuem espaço de memória reduzido e necessitando de leitores mais potentes. Usualmente são as que permitem apenas a leitura dos dados armazenados, não permitindo a gravação de dados extras (Patrick Sweeney, 2008).

As etiquetas RFID semi-passivas, possuem alimentação própria, porém não utilizam esta fonte de energia na transmissão do sinal de rádio. A fonte de alimentação é utilizada para o armazenamento de informação, processamento e qualquer outro tipo de atividade que exija uma fonte energética. Por exemplo, um produtor de alimentos pode aplicar estas etiquetas RFID semi-passivas para monitorar a temperatura de seu produto durante o transporte e o armazenamento. Quando a temperatura aumentar ou diminuir de certo ponto esta ocorrência pode ser armazenada na etiqueta automaticamente pelo sensor, para posterior verificação das corretas condições de armazenamento e transporte (Finkenzeller, 2009).

#### 3.1.4.2 – Leitor RFID

Um leitor RFID atua como uma ponte entre a etiqueta RFID, instalada no objeto, e o middleware e possui algumas funções básicas, tais como ler os dados armazenados na etiqueta RFID, transmitir os dados ao middleware e energizar a etiqueta, no caso de etiquetas RFID passivas.

Uma vez que o leitor se comunica com os identificadores usando rádio frequência, qualquer um deve ter uma ou mais antenas. Devido ao fato de um leitor comunicar-se com algum outro dispositivo ou servidor, o leitor também deve ter uma interface de rede de algum tipo. Exemplos de interfaces de rede comuns são a Universal Asynchronous Receiver/Transmitters (UARTs) serial para comunicações RS 232 ou RS 485 e o conector RJ45 para 10BaseT ou cabos 100BaseT Ethernet, alguns leitores possuem até mesmo Bluetooth ou comunicação Ethernet sem fio (Glover e Bhatt, 2007).



Figura 3.6 – Leitor RFID com Saída RS232 e ethernet (Shenzen Marktrace Co, 2010)

Basicamente os leitores são rádios, só que ao contrário dos rádios comuns, recebem como sinal ondas eletromagnéticas contendo as respostas dos identificadores com os dados armazenados. Podendo efetuar a leitura através de diversos materiais como papel, cimento, plástico, madeira, vidro e outros materiais. O leitor emite frequências de rádio em diversos sentidos no espaço, dependendo da antena e da frequência utilizadas, o alcance pode ir de alguns centímetros, até mesmo algumas dezenas de metros.

Existem ainda leitores RFID que possuem algumas características mais complexas de serem implementadas do que as apresentadas acima, como por exemplo, implementar medidas de anti-colisão da informação quando estão em comunicação com várias tags, autenticação de tags para evitar fraudes e acesso não autorizados e criptografia dos dados para garantir a integridade da informação (Glover e Bhatt, 2007).

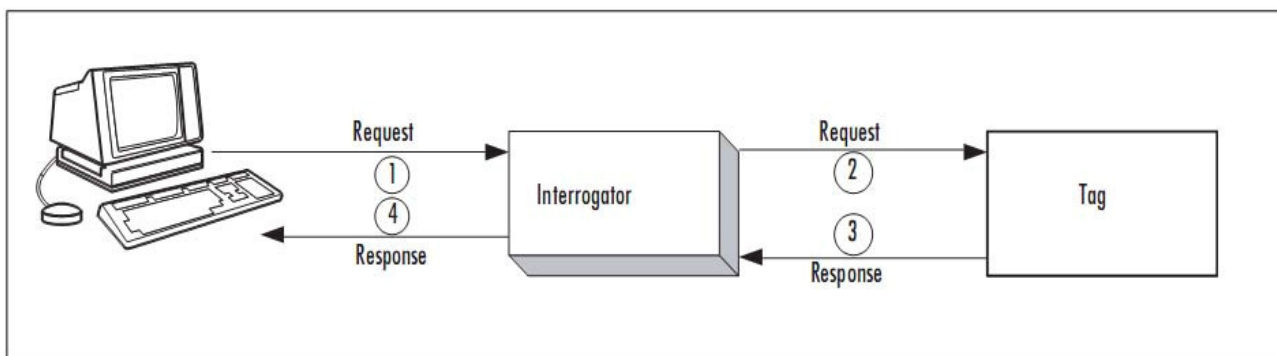


Figura 3.7 – Funcionamento de um leitor (Sweeney, 2008)

Algoritmos de anti-colisão são implementados para permitir que o leitor possa se comunicar com várias etiquetas RFID ao mesmo tempo. Existem três tipos de técnicas de anti-colisão utilizadas: espaciais, de frequência e de domínio de tempo. As três são utilizadas para estabelecer uma hierarquia de comunicação a fim de evitar que não se perca nenhuma informação das etiquetas que estejam no campo de identificação do leitor (John Wiley, 2007).

Em sistemas de alta segurança existe a necessidade de o leitor autenticar usuários, no caso os identificadores. A comunicação só se torna possível se os dois equipamentos (leitor e o identificador) trocarem uma senha em comum, previamente estabelecida.

A encriptação dos dados é uma medida de segurança para evitar ataques externos ao sistema. O leitor implementa a criptografia e a deciptografia dos dados para evitar ações fraudulentas e garantir a veracidade dos dados informados.

### 3.1.5 – Frequências de operação

As frequências operacionais são as frequências utilizadas para o envio do sinal eletromagnético que a etiqueta e o leitor RFID utilizam para realizar a comunicação entre si. A frequência de um sinal é a grandeza que indica a velocidade de repetição de um fenômeno periódico. Para transmissões de rádio, a frequência é uma das grandezas mais relevantes, pois é através dela que é possível estudar o comportamento dos sistemas em relação ao ambiente no qual o sistema será instalado. Devido ao fato de os sistemas RFID transmitirem ondas eletromagnéticas são regulados como dispositivo de rádio. Sistemas RFID não devem

interferir com outras aplicações protegidas, como rádios de serviços de emergência ou transmissão de televisão.

Geralmente o espectro eletromagnético na extensão na qual o RFID opera é dividido em frequência baixa, alta, ultra-alta a micro-ondas, como mostra a tabela 3.1 abaixo.

Tabela 3.1 – Faixas de Frequência RFID

<b>Nome</b>	<b>Faixa de Frequência</b>	<b>Frequência ISM</b>
LF (Frequência Baixa)	30-300 kHz	< 135 kHz
HF (Frequência Alta)	3-30 MHz	6,78 MHz, 13,56 MHz, 27,125 MHz, 40,680 MHz
UHF (Frequência Ultra-Alta)	300 MHz - 3 GHz	433,920 MHz, 869 MHz, 915 MHz
Micro-ondas	> 3 GHz	2,45 GHz, 5,8 GHz, 24,125 GHz

Fonte: Glover 2009

A tabela a seguir relaciona as frequências de onda com a faixa de leitura e apresenta o tipo de identificador que é usualmente utilizado.

Tabela 3.2 – Frequências, distância e tipo de etiqueta.

<b>Frequência</b>	<b>Faixa de Leitura</b>	<b>Tipo de Etiqueta</b>
LF	60 cm	Passiva
HF	1,5 - 3 metros	Passiva
UHF	3 - 9 metros	Ativa
Micro-ondas	> 9 metros	Ativa

Fonte: Glover 2009

No passado as aplicações com identificadores que operam nas faixas de LF e HF eram mais comuns. Com a redução nos custos de identificadores UHF passou-se a utilizar mais destes identificadores. No entanto o UHF não deverá substituir a LF em identificadores implantados em aplicações de extensão longa (faixa de leitura maior que 9 metros), em baixas frequências a faixa de leitura dos identificadores não passa de um metro ficando na casa dos 60 a 70 centímetros. Em altas frequências a faixa de leitura normalmente aumenta, especialmente quando são utilizadas etiquetas ativas. Tipicamente são utilizadas etiquetas passivas para operar nas faixas de baixa e alta frequência e etiquetas ativas para operar nas faixas de ultra-alta frequência e micro-ondas.

### 3.1.6 – Por que utilizar o RFID?

O código de barras provavelmente seja o identificador por computador mais familiar para os usuários atualmente. Porém o laser necessário para varrer o código de barras impõe uma limitação. É necessário uma “linha de visão” direta, de forma que o item tem que estar bem de frente e na direção correta, sem algo bloqueando o feixe entre o laser e o código de barra. A maior parte das outras formas de identificação, como tarjas magnéticas, biometria, reconhecimento ótico de caracteres e Smart Card, também devem se alinhar corretamente com o leitor ou ser inseridas de alguma forma específica.

Identificadores RFID fornecem um mecanismo para identificar um item a distância, com muito menos sensibilidade à orientação da etiqueta com o leitor, ou seja, não se faz necessário uma “linha de visão” entre a etiqueta e o leitor RFID. Possuem ainda a capacidade de identificar múltiplas etiquetas RFID ao mesmo tempo e identificar de forma única bilhões de itens, além de possuírem uma variedade de formas e a possibilidade de regravação, podendo reutilizar as etiquetas RFID.

## 3.2 – Linguagem de Programação – JAVA

### 3.2.1 – Introdução

A tecnologia *JAVA* começou a ser criada em 1991, devido a um projeto de pesquisa corporativa interna, financiado pela *SUN*, com o codinome *Green*. O projeto era esperado como a próxima geração de *software* embarcado. O resultado foi o desenvolvimento de uma linguagem baseada em *C++* que seu criador, *James Gosling*, chamou de *Oak*. Mais tarde descobriu-se que já havia uma linguagem de computador chamada *Oak*, foi quando uma equipe da *Sun* sugeriu o nome *Java* (Sierra e Bates, 2009).

O projeto *Green* passou por algumas dificuldades, devido ao fato de o mercado de dispositivos eletrônicos inteligentes voltado para o consumo popular não estava se desenvolvendo tão rápido quanto a *Sun* havia antecipado. Por uma feliz casualidade a *World Wide Web* explodiu em popularidade em 1993 e a equipe *Sun* viu o imediato potencial de utilizar o *Java* para adicionar conteúdos dinâmicos, como interatividade e animações, às páginas Web (Sierra e Bates, 2009).

A *Sun* anunciou o *Java* formalmente em uma importante conferência em maio de 1995. O *Java* chamou a atenção da comunidade de negócios por causa do enorme interesse na *World Wide Web*. O *Java* é agora utilizado para desenvolver aplicativos corporativos de grande porte, aprimorar a funcionalidade de servidores Web, fornecer aplicativos para dispositivos voltados para o consumo popular e para muitos outros propósitos.

### 3.2.2 – Conceitos sobre *Java*

#### 3.2.2.1 – Bibliotecas de classe do *Java*

Programas *Java* consistem em partes chamadas de classes. As classes incluem partes chamadas de métodos que realizam tarefas e retornam informações ao concluir. Os programadores podem criar cada parte de que precisam para formar programas *Java*, entretanto, a maioria dos programadores *Java* tira proveito das ricas coleções de classes



existentes nas bibliotecas de classe *Java*, que também são conhecidas *APIs do Java* ou *Java APIs (Application Programming Interfaces)*. Ao programar em *Java*, em geral, utilizamos os seguintes blocos de construção: classes e métodos de bibliotecas de classe, classe e métodos que você mesmo cria e classes e métodos que outros criam e tornam disponíveis (Sierra e Bates, 2009).

#### 3.2.2.2 – Ambiente de Desenvolvimento Java

Os programas Java, em geral, passam por cinco fases: edição, compilação, carga, verificação e execução.

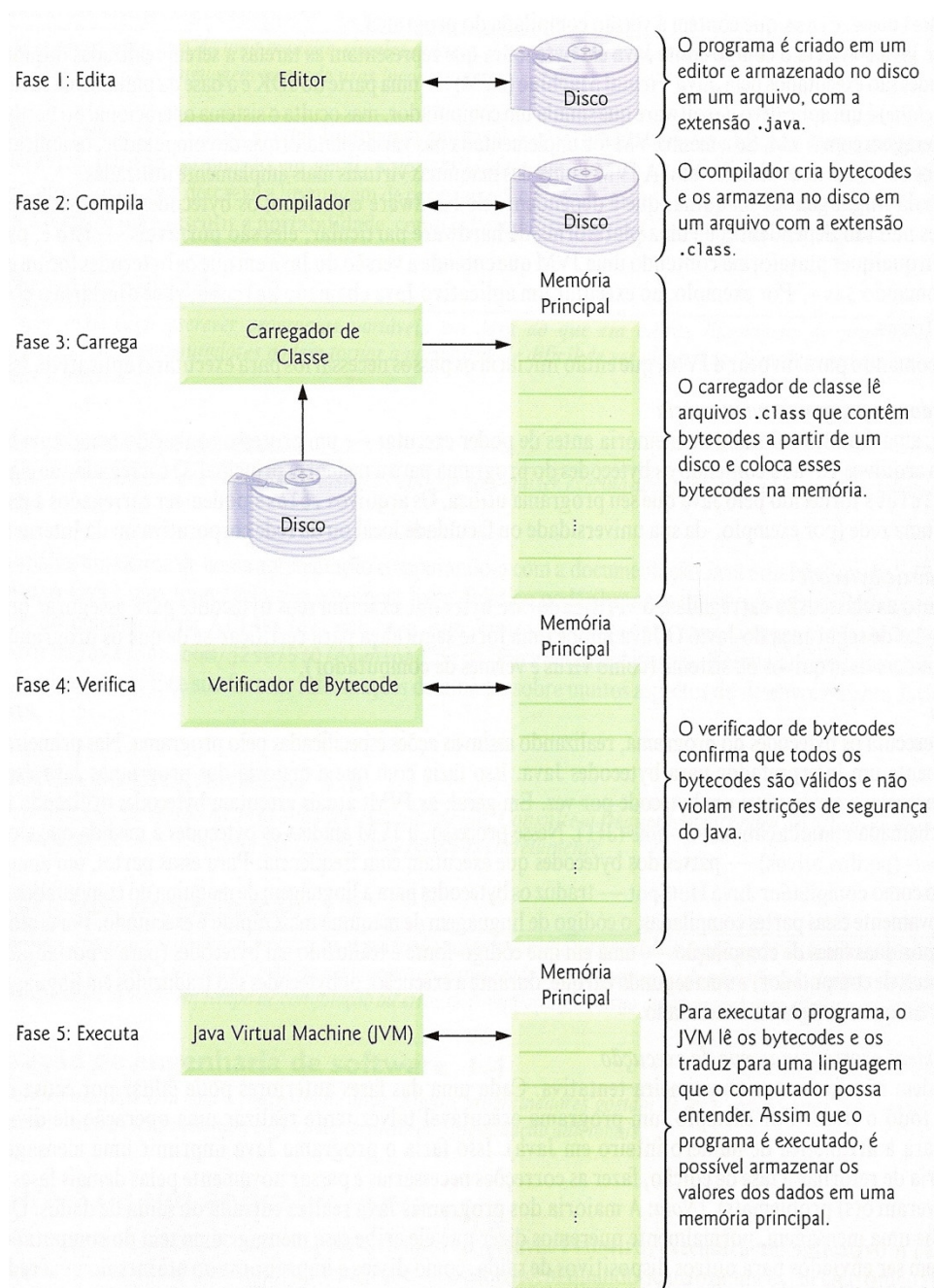


Figura 3.8 – Fases do programa em Java (Sierra e Bates, 2009)

A 1ª fase consiste em editar um arquivo com um programa editor. Normalmente digita-se o código-fonte utilizando um editor qualquer, faz quaisquer correções necessárias e salva o programa em um dispositivo de armazenamento secundário, como a unidade de disco rígido, por exemplo. Também existem os ambientes de desenvolvimento de software integrado (*IDEs – Integrated Development Enviorenents*) que estão disponíveis a partir de muitos fornecedores de *softwares*. Os *IDEs* fornecem ferramentas que suportam o processo de desenvolvimento de *software*, incluindo editores para escrever e editar programas e

depuradores para localizar erros de lógica em programas. Entre os vários *IDEs* populares, destacam-se o *NetBeans*, o *jEdit* e o *Eclipse* (Sierra e Bates, 2009).

Na 2ª fase o programador utiliza o comando *javac* (o compilador *Java*) para compilar um programa. Por exemplo, para compilar um programa chamado *Welcome.java*, você digitaria *javac Wlcome. Java* na janela de comando do sistema (*MS-DOS, Prompt de Comando, Prompt do Shell*). Se compilar, o compilador produz um arquivo *.class* chamado de *welcome.class* que contem a versão compilada do programa. O compilador *Java* converte o código fonte *Java* em *bytecodes* que representam as tarefas a serem realizadas durante a fase de execução. Os *bytecodes* são executados pela *JVM – Java virtual Machine* – uma parte do *JDK* e a base da plataforma *Java*. A máquina Virtual (*VM – Virtual Machine*) é um aplicativo de *software* que simula um computador, mas oculta o sistema operacional e o *hardware* subjacentes dos programas que interagem coma *VM*. Se a mesma *VM* for implementadas nas várias plataformas de computador, os aplicativos que ela executa podem ser utilizados em todas essas plataformas (Sierra e Bates, 2009).

Na 3ª fase o programa é carregado na memória antes de executar. O carregador de classe transfere os arquivos *.class* contendo os *bytecodes* do programa para a memória principal.

Na 4ª fase, enquanto as classes são carregadas, o verificador de *bytecode* examina os *bytecodes* para assegurar que eles são válidos e não violam restrições de segurança do *Java*.

Na 5ª fase, a *JVM* executa os *bytecodes* do programa, realizando assim as ações especificadas pelo programa. Em geral as *JVMs* atuais executam *bytecodes* utilizando uma combinação de interpretação e a chamada compilação *just-in-time (JIT)*. Nesse processo a *JVM* analisa os *bytecodes* à medida que eles são interpretados procurando *hot spots* (pontos ativos) – parte dos *bytecodes* que executam com frequência. Para essas partes, um compilador *just-in-time* – conhecido como compilador *Java HotSpot* – traduz os *bytecodes* para a linguagem de máquina subjacente.

### 3.3 – Banco de dados MySQL

#### 3.3.1 – Banco de dados

Um banco de dados é uma coleção de dados estruturados. Ele pode ser qualquer coisa desde uma simples lista de compras a uma galeria de imagens ou a grande quantidade de informação de uma rede corporativa. Para adicionar, acessar, e processar dados armazenados em um banco de dados de um computador, você necessita de um sistema de gerenciamento de bancos de dados como o Servidor *MySQL*. Como os computadores são muito bons em lidar com grandes quantidades de dados, o gerenciamento de bancos de dados funciona como a engrenagem central na computação, seja como utilitários independentes ou como partes de outras aplicações.

Um banco de dados relacional armazena dados em tabelas separadas em vez de colocar todos os dados num só local. Isso proporciona velocidade e flexibilidade. A linguagem *SQL* (*Structured Query Language* - Linguagem Estrutural de Consultas) é a linguagem padrão mais comum usada para acessar banco de dados e é definida pelo Padrão *ANSI/ISO SQL*.

#### 3.3.2 – *MySQL*

O software *MySQL* é um servidor robusto de bancos de dados *SQL* multitarefa e multiusuário. O Servidor *MySQL* pode ser usado em sistemas de produção com alta carga e missão crítica bem como pode ser embutido em programa de uso em massa. O programa *MySQL* é de Licença Dupla, ou seja, os usuários podem escolher entre usar o software *MySQL* como um produto *Open Source/Free Software* sob os termos da *GNU General Public License* (<http://www.fsf.org/licenses/>) ou podem comprar uma licença comercial padrão da *MySQL AB* (Milani, 2009).

O sistema gerenciador de banco de dados *MySQL* é um sistema cliente/servidor, que consiste de um servidor *SQL* multitarefa que suporta diferentes tipos de acessos, diversos

programas clientes e bibliotecas, ferramentas administrativas e diversas interfaces de programas.

Inicialmente o *MySQL* foi projetado para trabalhar com aplicações de pequeno a médio porte, algo em torno de 100 milhões de registros por tabela, tendo como tamanho médio aproximadamente 100 MB por tabela. Contudo, esses números eram os recomendados para as primeiras versões da ferramenta. Atualmente os limites e capacidades do *MySQL* ultrapassaram essas fronteiras inúmeras vezes. Hoje existem usuários que utilizam o Servidor *MySQL* com 60.000 tabelas e aproximadamente 5.000.000.000 de linhas.

A maioria dos sistemas operacionais existentes no mercado suporta a execução do *MySQL*. Por ser um programa escrito em *C* e *C++*, isto torna extremamente fácil a sua portabilidade entre diferentes plataformas, como por exemplo: *Linux* (*Fedora Core*, *Debian*, *SuSE*, *RedHat*), *Unix* (*Solaris*, *HP-UX*, *AIX*, *SCO*), *FreeBSD*, *Mac OS X Server* e *Windows*. As plataformas mais indicadas para o uso com o *MySQL* são aquelas que possuem a biblioteca *thread* e o gerenciador de arquivos mais estáveis.

### 3.3.2.1 – Principais características

Boa parte do *MySQL* apoia-se na linguagem *SQL* e suporta um subconjunto estendido do padrão *ANSI SQL*. O *MySQL* é um servidor *mutli-thread*, isto é, cria múltiplos processos de execução em um único processo para atender a diversos pedidos de usuários ao mesmo tempo. Ele executa tarefas *back-end* (não controladas pelo usuário) e reúne vários programas e bibliotecas de cliente, ferramentas administrativas e *APIs*, além de vários aplicativos desenvolvidos por colaboradores e parceiros.

O *MySQL* foi desenhado em várias camadas e entre suas principais características pode-se destacar:

- Replicação para grandes servidores;
- Manipular tabelas *InnoDB* para sistemas pesados. *InnoDB* é um tipo de tabela nativa do *MySQL*, constituída de um arquivo de cabeçalho e de um “*tablespace*”, semelhante ao do Oracle, em que os dados são armazenados.
- *MyODBC*, uma implementação do conector *ODBC* para *MySQL* no acesso a bases de dados do *Windows*;

### 3.3.2.2 - Arquitetura

O *MySQL* é um gerenciador constituído em múltiplas camadas que podem ser classificadas basicamente em camada de apresentação, de lógica e física.

A camada de apresentação é constituída pelas interfaces de usuários, isto é, a forma como o servidor interage com seus usuários e pode ser dividida em três componentes: interface de consulta, cliente, e administrador.

A camada lógica representa as funcionalidades desenvolvidas pelo fornecedor. Elas podem ser agrupadas em três subsistemas principais, processador de consultas, gerenciamento de transições e gerenciamento de recuperação.

A camada física corresponde a forma como o *MySQL* mantém o armazenamento físico dos dados em arquivos de dados, dicionários, índices, históricos e estatísticas, além do controle dos recursos de memória e é composta por gerenciador de recursos, gerenciador de buffers e gerenciador de armazenamento.

## **CAPÍTULO 4 – SISTEMA PROPOSTO**

Este capítulo apresenta a implementação da solução ao problema exposto no capítulo 2. Aqui são detalhados os dispositivos eletrônicos, o banco de dados para armazenar as informações necessárias e o software desenvolvido para realizar o controle de todo o projeto, além de detalhar todas as etapas do desenvolvimento e apresentar a proposta de solução e sua implementação.

### **4.1 – Apresentação Geral do Sistema Proposto**

O modelo proposto pelo projeto é um sistema capaz de identificar um veículo sem a necessidade de intervenção humana, parando ou visualizando os veículos que trafegam em determinado ponto da via. O modelo final apresentado será um protótipo, com fins acadêmicos que poderá servir de para projetos comerciais futuros.

Para automatizar a identificação de veículos é utilizado a tecnologia RFID, com leitores e etiquetas RFID disponíveis no mercado, linguagem de programação JAVA e banco de dados MySQL. O projeto busca desenvolver uma ferramenta de software integrada a tecnologia RFID para identificar de forma única todos os veículos que passaram em determinado ponto da via e armazenar o histórico do ponto da via controlado.

Para apresentar uma proposta de sistema de monitoramento, inicialmente, foram determinadas quais as informações mínimas que deveriam existir para que fosse possível identificar, de forma única, todos os veículos utilizados no protótipo. Assim foram definidas que as seguintes informações do veículo seriam utilizadas:

- O chassi que é composto por 17 caracteres alfanuméricos que identificam os veículos automotores em geral;
- O RENAVAM que é o Registro Nacional de Veículos Automotores, onde cada veículo possui o próprio;
- A placa de identificação que possui 7 caracteres individualizados para cada veículo;
- O número da etiqueta RFID anexa ao veículo.

- A data e a hora que o veículo passou diante do leitor.

Como só identificar o veículo não satisfaz todas as necessidades encontradas nos dias atuais, o protótipo proposto permite que seja identificado, também, o proprietário do veículo, além de permitir a atualização dos dados do proprietário para que seja possível a atualização da base de dados.

Devido ao fato de ser um projeto acadêmico, sem fins comerciais e com restrições orçamentárias e de tempo, o protótipo apresentado do sistema de identificação de veículos não se preocupa com alguns aspectos, entre eles, a identificação de infrações de trânsito, possíveis congestionamentos na via monitorada e a identificação do condutor do veículo.

Como o dado que é transmitido entre a etiqueta RFID e o leitor é apenas o identificador único de cada veículo e toda a informação relativa ao veículo é armazenada em uma central de dados, não houve a preocupação com a criptografia dos dados na transmissão. Em um eventual ataque ao sistema de RFID a captura do identificador único do veículo, não surtiria efeitos desastrosos, já que com apenas essa informação não é possível adulterar os restantes dos dados do veículo.

## 4.2 – Descrição das Etapas do Sistema



Figura 4.1 – Etapas do Modelo

Fonte: Autor

O projeto foi desenvolvido em cinco etapas distintas. A primeira etapa foi identificar o problema e as soluções existentes para, em seguida, apresentar uma proposta de projeto e assim iniciar com o estudo bibliográfico. A segunda etapa foi marcada pelo o início do



desenvolvimento da monografia e por identificar as ferramentas e tecnologias necessárias para a montagem do protótipo. A terceira etapa se iniciou com o desenvolvimento do projeto e os primeiros testes necessários, que serão mostrados ainda neste capítulo, e finalizou com a montagem da maquete. A quarta etapa foi a conclusão do desenvolvimento da monografia, sua revisão final e testes finais no protótipo. Por fim, a quinta etapa será a montagem de maquete e apresentação na banca examinadora. A figura 4.1 mostra essa cinco etapas.

### 4.3 – Descrição da Implementação

Devido o fator acadêmico do projeto e a limitação de tempo para o desenvolvimento do projeto e da monografia, foi focado o desenvolvimento de um protótipo, que venha a fornecer a base do sistema de monitoramento e controle de uma via através de um cadastro básico de funcionalidades.

Os dispositivos RFID utilizados, tanto o leitor quanto a etiqueta que irá identificar o veículo, são controlados e monitorados por um sistema de gerenciamento que foi desenvolvido em linguagem *JAVA*, utilizando o software *NetBeans IDE 6.9.1*. Para o desenvolvimento do banco de dados, onde serão armazenados as características de cada veículo, foi utilizado o sistema gerenciador de banco de dados *MySQL*.

Na implementação da solução apresentada podemos verificar as seguintes tecnologias, o leitor e as etiquetas RFID, o banco de dados desenvolvido para aplicação e o sistema desenvolvido em Java para gerenciar a aplicação.

#### 4.3.1 – *Hardware* utilizado

##### 4.3.1.1 – O leitor RFID

A escolha do leitor RFID a ser utilizado foi uma etapa importante no projeto. Hoje existem diversos modelos disponíveis no mercado, porém todos os leitores RFID encontrados no início eram caros e muito complexos para o projeto. Após reuniões com professores e

colegas de faculdade o leitor RFID da marca canadense *Phidget* foi escolhido para ser utilizado no projeto.

O diferencial do leitor é o seu baixo custo, cerca de R\$ 70,00 (apenas o leitor) e a forma de comunicação do leitor com o computador, neste caso pela porta *USB – Universal Serial Bus*. A comunicação via porta *USB* facilita a implementação, pois não se faz necessário utilizar o padrão *RS-232* que utiliza a porta serial do computador.

Uma das características mais relevantes desse leitor é a ampla gama de interfaces para programação de aplicações (*Application Programming Interfaces - API*) oferecidas pelo fabricante. Dentre todas as opções de linguagem oferecidas entre elas *C#*, *C/C++*, *Flash AS3*, *Delphi* e *Visual Basic 6.0*, a linguagem *Java* foi selecionada. A figura abaixo mostra o leitor *Phidget*.



Figura 4.2 – Leitor *Phidget* RFID (*Phidget*, 2011)

Com as especificações do leitor, na tabela 4.1, pode-se identificar para quais aplicações o leitor pode ser utilizado. Devido a sua característica de pouca potência do sinal de saída da antena e a baixa frequência de ressonância do sinal do leitor, fica justificada as curtas distâncias de leitura das etiquetas. Isso explica do porque o leitor *PhidgetRFID* ser utilizado apenas como ferramenta acadêmica e para o desenvolvimento de pequenos projetos.

Tabela 4.1 – Características do Leitor RFID

<i>Antenna Output Power (max, far field)</i>	<i>&lt; 10 <math>\mu</math>W</i>
<i>Antenna Resonant Frequency</i>	<i>125kHz - 140kHz</i>
<i>Communication Protocol</i>	<i>EM4102</i>
<i>Read Update Rate</i>	<i>30 updates / second</i>
<i>External +5V Supply Voltage</i>	<i>5VDC</i>
<i>External LED Supply Voltage</i>	<i>5VDC</i>
<i>External LED Supply Current Limit</i>	<i>16mA</i>
<i>External LED Output Resistance</i>	<i>250 Ohms</i>
<i>Recommended Terminal Wire Size</i>	<i>16 - 26 AWG</i>
<i>Device Quiescent Current Consumption</i>	<i>16mA</i>
<i>Device Active Current Consumption</i>	<i>100mA max</i>
<i>Typical Read Distance - Credit Card Tag</i>	<i>11cm (5")</i>
<i>Typical Read Distance - Disk Tag</i>	<i>6cm (3")</i>
<i>Typical Read Distance - Key Fob Tag</i>	<i>7cm (3.5")</i>

Fonte: *Phidget*, 2011

Para um leitor *RFID*, como o utilizado no projeto, se comunicar com as etiquetas *RFID*, os dois dispositivos devem compartilhar um mesmo protocolo de comunicação. O protocolo atua como um grupo de regras para definir a forma de comunicação sem fio entre o leitor e a etiqueta. O leitor e as etiquetas *RFID* da *Phidget* utilizam o protocolo *EM4102*. Assim quaisquer outras etiquetas *RFID* que utilizam o mesmo protocolo podem ser lidas pelo leitor.

Levando em conta que o objetivo final do projeto é o desenvolvimento de um protótipo, foram utilizados como limites as características do leitor *RFID* em uso no projeto.

Como principais características técnicas do leitor *Phidget*, podemos destacar que lê etiquetas *RFID* trazidas no mínimo 7 cm do leitor, identifica qualquer etiqueta que utilize o protocolo

*EM4102*, retorna o *ID* único contido em cada etiqueta, provê duas saídas digitais para controlar *LEDs*, relays, contém um *LED* e conecta-se diretamente a USB do computador.

Como as etiquetas *RFID* passivas necessitam de um forte campo de Rádio Frequência para operar, seu limite de leitura é bem menor que as etiquetas ativas. O campo de alcance proporcionado pelo leitor é onde as etiquetas passivas funcionam. No caso do leitor *Phidget*, as etiquetas aproximadas em até 7 cm de distancia do leitor podem ser identificadas. Ultrapassando esse limite o leitor já não consegue mais identificar a etiqueta *RFID*.

Outra limitação encontrada foi a incapacidade de identificar múltiplas etiquetas *RFID* quando todas estão na zona de identificação do leitor. Ou seja, caso duas ou mais etiquetas entrem no campo de identificação do leitor, ao mesmo tempo nenhuma etiqueta será lida. Dessa forma uma etiqueta deverá ser removida.

Levando em conta que o projeto final é um protótipo, a distância de máxima de alcance de 7 cm do leitor e a impossibilidade de identificar duas etiquetas no campo de leitura, foram tratadas como exceções, pois a simulação terá como limites principais estas duas características presentes no leitor.

O protocolo *EM4102* é um circuito integrado para uso em eletrônicos que utilizam a identificação por radiofrequência. O circuito é acionado por uma bobina que gera uma campo eletromagnético e seu clock é o mesmo de terminais externos.

#### 4.3.1.2 – Teste do Leitor *RFID*

A instalação do *software*, *Phidget21 Installer*, que deve ser instalado para testar o leitor *RFID*, executa a instalação de uma biblioteca nativa no Windows para efetuar a comunicação do sistema operacional com o leitor acoplado via porta USB, como mostra a imagem abaixo, e de um programa de configuração e teste do leitor. Após a instalação é possível verificar se o leitor está funcionando corretamente, testando a comunicação dele com uma etiqueta *RFID*.



Figura 4.3 - Leitor RFID acoplado no computador

Fonte: Autor

Através do painel de controle do sistema, como mostrado na figura abaixo, podemos visualizar que o leitor foi identificado com sucesso pelo sistema operacional.

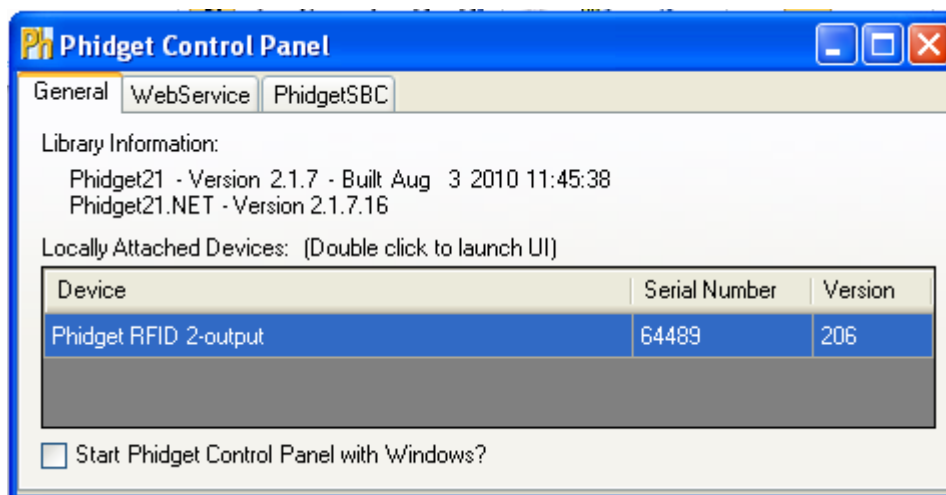


Figura 4.4 - Painel de Controle Phidget

Fonte: Autor

Com a identificação do leitor Phidget, foi possível iniciar os primeiros testes com o leitor RFID. Na imagem abaixo podemos verificar o teste do leitor com a aproximação da etiqueta RFID 19007b1550.

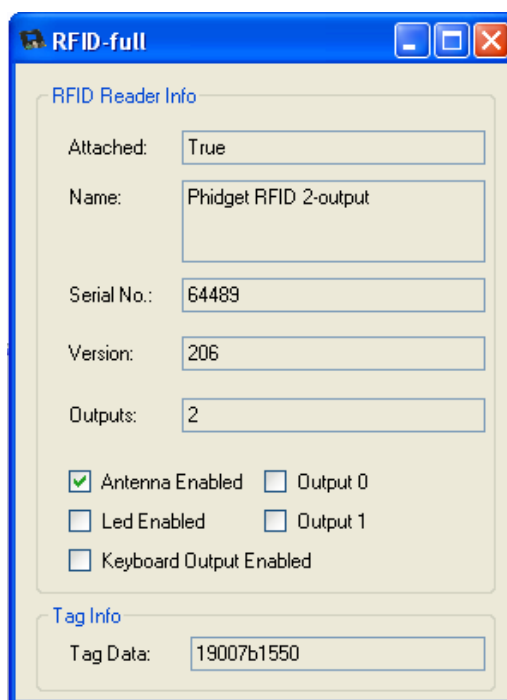


Figura 4.5 – Teste Leitor RFID

Fonte: Autor

#### 4.3.1.3 – Etiquetas RFID

As etiquetas RFID utilizadas no projeto também são da marca canadense Phidget. Possuem a características de serem etiquetas passivas, ou seja, não possuem uma fonte interna de alimentação, nesse caso, a etiqueta utiliza do campo magnético gerado pelo próprio leitor RFID para poder responder ao leitor RFID.

A informação contida na etiqueta RFID já vem gravada de fabrica e não é possível realizar alterações ou adicionar qualquer tipo de informação na etiqueta, dessa forma, foi necessário utilizar um banco de dados para o armazenamento dos dados e um sistema de cadastro, que será detalhado mais a frente.

#### 4.3.1.4 – O Protocolo de comunicação

Para um leitor RFID (como o RFID – Phidget) se comunicar com as etiquetas RFID, eles devem compartilhar um mesmo protocolo de comunicação. Esse protocolo atua como um grupo de regras para definir a forma de comunicação sem fio entre o leitor e a etiqueta. O leitor RFID – Phidget (assim como as etiquetas vendidas pela Phidget) utilizam o protocolo EM4102, que utiliza uma frequência de operação de 125 KHz. Quaisquer outras etiquetas que também utilizem o protocolo EM4102 podem ser lidas pelo leitor RFID – Phidget.

O protocolo EM4102 contém 64 bits divididos em cinco grupos de informação. Nove bits para serem usados como cabeçalhos, 10 bits de paridade de linha, 4 bits utilizados para a paridade de coluna, 40 bits para dados e 1 bit utilizado como final da informação.

#### 4.3.2 – Metodologias e ferramentas utilizadas.

Todas as ferramentas utilizadas na implantação desse projeto são gratuitas. O ambiente de desenvolvimento escolhido para desenvolver o *software* e a interface gráfica do projeto foi o *NetBeans 6.9.1*. Para desenvolver o banco de dados necessário escolheu-se o Sistema Gerenciador de Banco de Dados *MySQL* pela experiência anterior na matéria de banco de dados. Para a modelagem do banco de dados foi utilizado o *MySQL Workbench* que auxilia na criação de tabelas e relacionamentos. Para a modelagem da maquete que será utilizada na apresentação para banca foi utilizado o *Google Sketchup*. A figura abaixo mostra todas as ferramentas utilizadas.



Figura 4.6 – Ferramentas Utilizadas

Fonte: Autor

Todo o *software* do projeto foi inteiramente desenvolvido na linguagem de programação *Java*. Sua utilização deve-se à anterior experiência com a linguagem de programação e a grande variedade de material disponível na internet e em livros. Além disso, o leitor RFID, possui uma interface de programação de aplicativos disponibilizada em Java o que facilitou bastante a comunicação e a integração do hardware com o software.

#### 4.3.2.1 – Diagrama de *Software*

Como metodologia de desenvolvimento, o sistema foi dividido em sete pacotes de classe. Cada pacote contém classes afins que exercem funções parecidas. Os pacotes são listados a seguir: *ListenersSC*, *ListenersSM*, *MiddlewareRFID*, *SISMOVE.Conexao*, *SISMOVE.Forms*, *SISMOVE.Logica* e *SISMOVE.SQL*. A figura 4.7 mostra todos os pacotes utilizados para a programação do sistema.

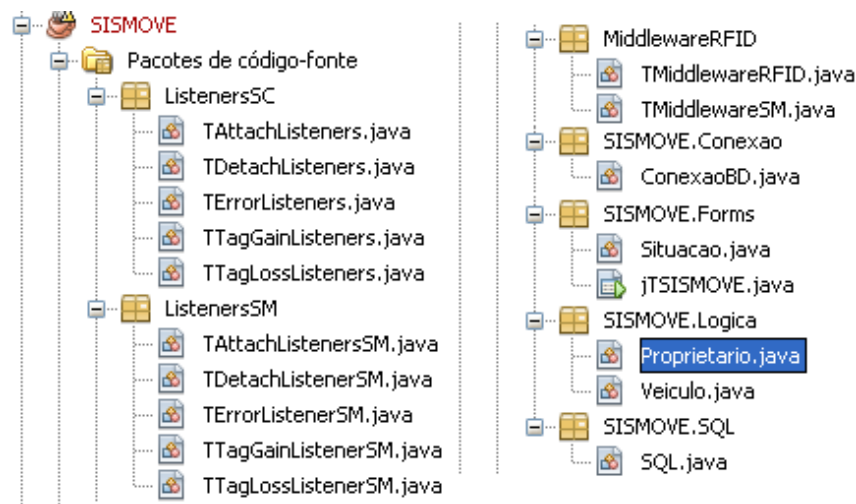


Figura 4.7 – Pacotes utilizados no desenvolvimento do software.

Fonte: Autor



#### 4.3.2.1.1 - Pacote “SISMOVE.Forms”

Esse pacote contém a classe principal do sistema. É aqui que foi desenvolvido a classe “jTSISMOVE.java” responsável por construir a interface gráfica do software. Nesta classe que se encontra todas as chamadas de funções que existe no software.

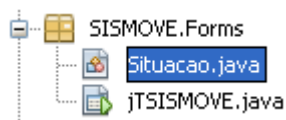


Figura 4.8 – Pacote “SISMOVE.Forms”

Fonte: Autor

#### 4.3.2.1.2 – Pacote “SISMOVE.Logica”

Neste pacote está a classe Proprietario.java e Veiculo.java. As duas classes foram desenvolvidas para conter o modelo para os objetos Proprietario e Veículo utilizados no software.

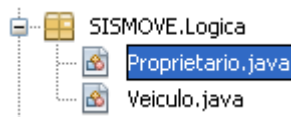


Figura 4.9 – Pacote SISMOVE.Logica

Fonte: Autor

#### 4.3.2.1.3– Pacote “SISMOVE.Conexao e SISMOVE.SQL”

Os dois pacotes apresentados abaixo foram desenvolvidas para realizarem o acesso ao banco de dados. O pacote “SISMOVE.Conexão” possui a classe `ConexaoBD.java`, que possui o código responsável por conectar o software ao banco de dados. Já o pacote “SISMOVE.SQL” possui a classe “`SQL.java`”, responsável por alterar, inserir, apagar e consultar os registros das tabelas no banco de dados.

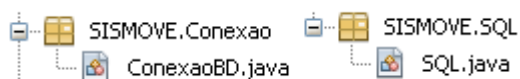


Figura 4.10 – Pacotes “SISMOVE.Conexao” e “SISMOVE.SQL”.

Fonte: Autor

#### 4.3.2.1.4 - Pacote “MiddlewareRFID”

Neste pacote são agrupadas as duas classes necessárias que possuem funcionamento similar a de um middleware. Essas classes fazem a comunicação entre o hardware (o leitor RFID) e as demais classes que utilizam as funcionalidades do hardware. As duas classes utilizadas são: `TMiddlewareRFID.java` e `TMiddlewareSM.java`.



Figura 4.11 – Pacote MiddlewareRFID.

Fonte: Autor

#### 4.3.2.1.5 – Pacote “ListenersSC e ListenersSM”

Os pacotes “ListenersSC” e “ListenersSM” são classes que tratam eventos providos do hardware (leitor RFID). O pacote “ListenersSC” trata eventos do leitor RFID no momento

que está sendo realizado o cadastrado de um novo veículo. O pacote “ListenersSM” trata eventos quando se está monitorando uma via. Os dois pacotes possuem as seguinte classes: AttachListeners, DettachListeners, ErrorListeners, TagGainListeners, TagLossListeners.

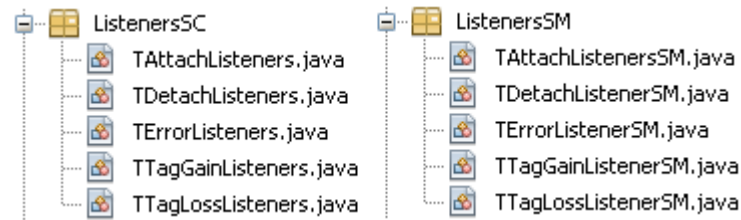


Figura 4.12 – Pacotes “ListenersSC” e “ListenersSM”

Fonte: Autor

#### 4.3.2.2 - Diagrama de dados

Abaixo, a figura 4.13, apresenta todas as tabelas que compõe o banco de dados utilizado no projeto.

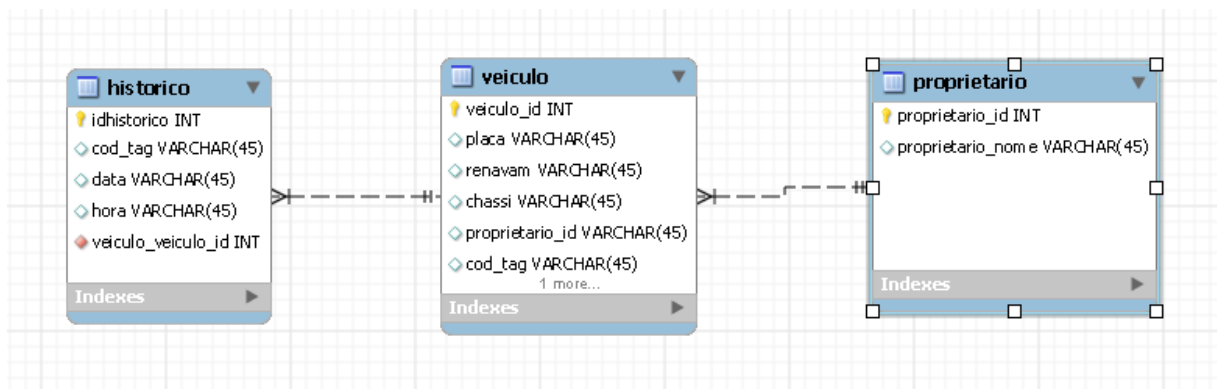


Figura 4.13 – Banco de dados utilizado

Fonte: Autor

Primeiramente temos a tabela veículo que armazena as informações sobre a identificação do veículo (Chassi, RENAVAM e placa) perante o órgão executivo de trânsito e possui os seguintes campos:

- Veiculo\_id que guarda o identificador único de cada veículo e é a chave primária da tabela, aqui quem escolhe o identificador, a ser utilizado é o operador do sistema.
- Placa: Armazena a placa de identificação, no formato LLL-NNNN, de cada veículo cadastrado.
- Renavam: Guarda o número do RENAVAM (Registro Nacional de Veículos Automotores) do veículo cadastrado.
- Chassi: armazena o número do CHASSI de cada veículo.
- Proprietário: que armazena o nome do proprietário do veículo
- Cod\_tag: onde é armazenado o identificador único de cada veículo, ou seja, o identificador da etiqueta RFID que está anexada ao veículo.

A tabela proprietário armazena os dados necessários para identificar os proprietários de cada veículo. E possui os seguintes campos para identificar cada dono:

- Proprietário\_id: Armazena um número de identificação de cada proprietário, podendo ser o CPF (cadastro de pessoa física) do cidadão.
- Nome: Guarda o nome completo de cada proprietário;

Para armazenar o histórico da via foi criado a tabela *histórico* que guarda as informações da leitura realizado pelo leitor RFID. Dessa forma essa tabela possui os seguintes campos:

- idhistorico: para ser utilizado como chave primária quando for necessário recuperar algum dado;
- Data: para armazenar a data da leitura do veículo;
- Hora: para armazenar o horário que o leitor RFID realizou a identificação do veículo; e
- Cod\_tag: para identificar qual veículo é passou em determinado dia e horário.

## **CAPÍTULO 5 – APLICAÇÃO DO MODELO PROPOSTO**

Neste capítulo será detalhado a área de aplicação do modelo proposto, bem como, a apresentação do modelo envolvendo um caso real. O objetivo é mostrar a viabilidade da proposta, bem como mostrar que o modelo proposto pode servir de base para novos projetos.

Não será apresentado um projeto definitivo que estará apto a ser utilizado. Será demonstrado como é possível monitorar, planejar e melhorar a gestão do trânsito através da identificação eletrônica dos veículos utilizando RFID.

### **5.1 - Apresentação da área de Aplicação do Modelo Proposto**

A área de aplicação do modelo proposto, como já foi dito, é o trânsito. Hoje no país existe o Código Brasileiro de Trânsito – CTB, lei nº 9.503 de setembro de 1997, que define as atribuições das diversas autoridades e órgãos ligados ao trânsito. É o CTB que fornece as diretrizes para a engenharia de tráfego, estabelece normas de conduta, infrações e penalidades.

É definido que o trânsito em condições seguras é um direito de todos e dever dos órgãos e entidades do Sistema Nacional de Trânsito. Cabe a estes órgãos adotar medidas que assegurem este direito e direcionem as ações para melhorar o trânsito.

O Sistema Nacional de Trânsito é composto pelos seguintes órgãos e entidades:

- O Conselho Nacional de Trânsito – CONTRAN, coordenador do sistema e órgão máximo normativo e consultivo.
- Os Conselhos Estaduais de Trânsito – CETRAN, órgão normativo, consultivo e coordenador no âmbito dos Estados.
- Os órgãos e entidades executivos de trânsito da União, dos Estados, do Distrito Federal e dos Municípios;
- Os órgãos e entidades executivos rodoviários da União, dos Estados, do Distrito Federal e dos Municípios;
- A Polícia Rodoviária Federal;

- As Polícias Militares dos Estados e Municípios; e
- As juntas Administrativas de recursos de Infrações - JARI.

Utilizando como base a resolução nº 212 de 13 de dezembro de 2006 do CONTRAN, que institui o SINIAV – Sistema Nacional de Identificação Automática de Veículos, o modelo proposto se enquadra na necessidade de dotar os órgãos executivos de trânsito de instrumentos modernos e interoperáveis para monitorar, fiscalizar e gerir o trânsito e a frota de veículos nacional.

Dessa forma o projeto foi baseado em uma resolução expedida pelo CONTRAN, para justificar sua importância de ser implementado, pois como foi mostrado o transporte em rodovias e vias urbanas no país tende apenas a aumentar com o tempo, fazendo com que o monitoramento, a fiscalização e a gestão de forma eficiente se torne algo complexo e difícil para a segurança da frota de veículos.

Como se trata de informações que são geridas pelos Departamentos Estaduais e Nacional de Trânsito, os órgãos do Sistema Nacional de Trânsito ficariam incumbidos de gerir toda a implantação do sistema, como ocorre hoje com o SINIAV.

O resultado final do projeto busca demonstrar os ganhos de eficiência quando se é utilizado a tecnologia RFID para identificar veículos.

#### 5.1.1 - O ambiente de simulação

O ambiente de simulação de todas as funcionalidades do projeto final será a maquete produzida para tal fim. Nela será demonstrada uma via pública sendo monitorada pelo leitor RFID, onde todos os veículos presentes na simulação terão uma etiqueta RFID, com um número de série único. A imagem a baixo ilustra a onde será apresentado a simulação do projeto.

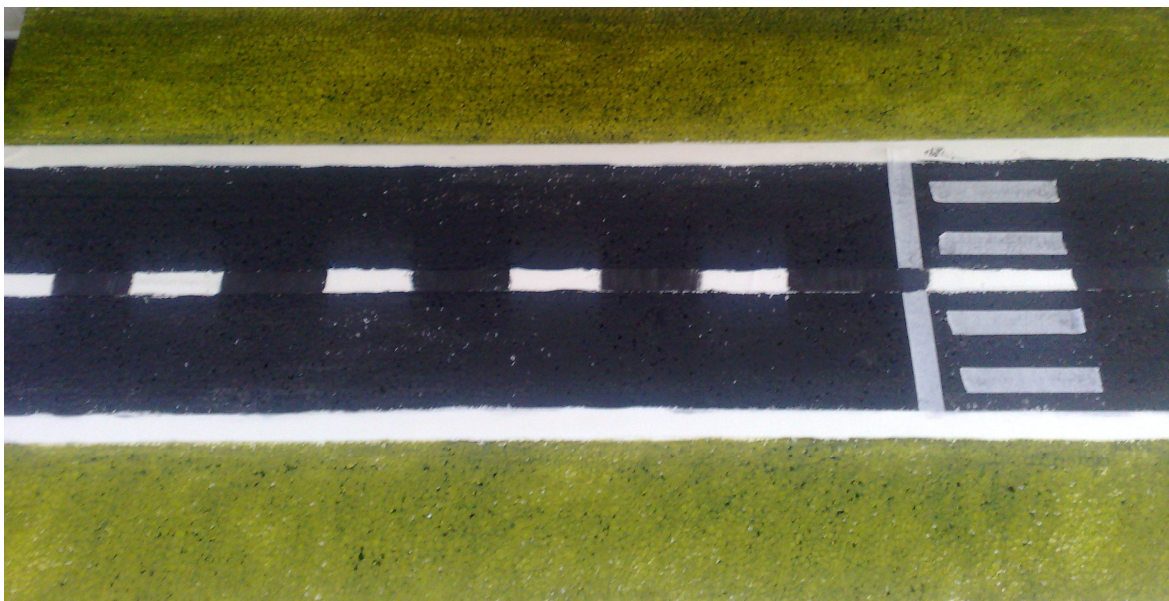


Figura 5.1 – Ilustração da Maquete utilizada.

Fonte: Autor

Para demonstrar o monitoramento da via, serão utilizados três veículos previamente cadastrados e um veículo que será cadastrado no momento da apresentação. A identificação do veículo em trânsito pela via se dá no momento em que o mesmo passa diante do leitor RFID. Dessa forma os dados do veículo são apresentados na tela do software desenvolvido.

## 5.2 - Descrição da Aplicação do Modelo

Nessa seção efetuar-se-á uma descrição detalhada de cada etapa necessária para o desenvolvimento do Sistema de Monitoramento de Veículos Automotores. Utilizando a tecnologia RFID. Ou seja, será feito um detalhamento do trabalho realizado para o desenvolvimento do sistema.

O sistema foi dividido em três módulos distintos. Cada módulo realiza uma função em separado, mas todos de certa forma são interligados. São eles: o módulo de cadastro, módulo de monitoramento e módulo de controle.

Nas seções subsequentes serão explicados módulo por módulo, justificando o porquê de sua existência e demonstrando a utilização dos métodos e técnicas escolhidos para sua programação.

### 5.2.1 - Módulo de cadastro

Esse módulo é essencial para o funcionamento do sistema como um todo, pois ele é responsável pelo cadastramento dos veículos utilizados no protótipo. Todas as informações do veículo são gravadas na tabela veículo. As informações do veículo e do proprietário do veículo são gravadas na tabela veículo e proprietário do banco de dados *MySQL*.

Para realizar o cadastramento de um novo veículo, primeiramente é necessário ativar o leitor RFID em seguida aproximar o veículo ou a etiqueta que será anexada ao veículo do leitor para que seja feita identificação da mesma. Em seguida, clicando em “Novo” serão habilitados os seguintes campos:

- ID\_Veículo: Deverá ser utilizado como se fosse uma chave primária. Ou seja, cada veículo deverá possuir um ID único que o identifique além da etiqueta RFID.
- Placa: Composta por três letras e quatro números além do símbolo “-” que os separam.
- Renavam: Composto por x caracteres, sendo o Registro Nacional do Veículo.
- Chassi: Combinação de dezessete caracteres alfanuméricos que identificam os veículos automotores em geral.
- Proprietário: O nome do proprietário do veículo sem abreviações.

Após o preenchimento de todos os dados e clicando em “Salvar” o *software* conecta-se com o banco de dados e insere todos os dados preenchidos e mais o identificador da etiqueta RFID na tabela veículo, gerando mais um registro.

A lista de *TAG's* mostra todas as etiquetas *RFID* cadastradas naquele momento. Se por acaso tentarmos cadastrar uma etiqueta *RFID* que já foi lida anteriormente o sistema gera uma mensagem de erro.

A imagem abaixo ilustra o módulo de cadastro do sistema.



Figura 5.2 – Cadastro de Veículos

Fonte: Autor

Como o módulo utiliza o hardware para seu funcionamento, ele faz uso de uma classe *middleware* e de seus *listeners* para a interação com o leitor RFID.

A classe *TMiddleware* possui quatro métodos que interagem com o leitor RFID. Esses métodos são utilizados por outras classes que desejam fazer uso do leitor RFID. São eles:

- *startRFID* que inicia os *listeners* e abre a comunicação com o leitor.
- *stopRFID* que para os *listeners* e fecha a comunicação com leitor.

Cada *listeners* possui uma função específica. O *TAttachListeners* gera eventos sempre que algum leitor RFID é conectado a alguma porta USB do computador, apresentando o nome do RFID no campo “Leitor Conectado” da imagem a cima. O *TDetachListeners* gera eventos sempre que o leitor RFID é desconectado da USB. O *TErrorListeners* gera eventos sempre há erro no leitor RFID. O *listener TTagGainListeners* gera eventos sempre que alguma etiqueta *RFID* é identificado pelo leitor e o *TTagLossListeners* gera eventos sempre que a etiqueta *RFID* sai do campo de identificação do leitor.

### 5.2.2 - Módulo de monitoramento

Esse módulo é o principal módulo do sistema de monitoramento proposto. É aqui que é realizado o monitoramento da via propriamente dito. É neste módulo que é apresentado todos os veículos que passaram diante do leitor *RFID* instalado na via.

O módulo de monitoramento interage com o leitor *RFID* e faz uma chamada no banco de dados para buscar as informações da etiqueta lida e apresenta nos campos específicos de cada informação.

Para que seja armazenado o histórico de veículos que circularam na via monitorada pelo leitor *RFID*, a placa de identificação dos veículos é armazenada juntamente com as informações de data e hora da passagem na tabela histórico do banco de dados.

A imagem abaixo ilustra o módulo de monitoramento.

Sistema de Monitoramento e Controle

Leitor RFID

Ativar Desativar

Leitor Conectado:

Placa dos Veículos na Vi

Hora da Passagem

Data da Passagem

Veiculo

Tag ID: Chassi: Renavam: Placa: Proprietário:

Figura 5.3 – Módulo de Monitoramento.

Fonte: Autor

Da mesma forma do módulo de cadastro, o módulo de monitoramento também interage com o *hardware*. Por isso, ele possui uma classe de *middleware* (*TMiddlewareSM*) e seus respectivos *listeners* para fazer a interação com o leitor *RFID*. O funcionamento da

classe middleware e seus *listeners* são semelhantes ao do sistema cadastro. A diferença está na codificação interna de cada *listener*.

### 5.2.3 - Módulo de controle

O módulo de controle permite verificar todos os veículos que existem cadastrados no banco de dados com seus respectivos dados e identificador da etiqueta RFID. É neste módulo que é possível alterar os dados de um veículo ou excluir o registro do veículo da base do banco de dados. Dessa forma assim que este veículo passar diante do leitor RFID o sistema de monitoramento irá gerar uma mensagem informando que determinado veículo não se encontra na base de dados. A imagem abaixo apresenta o módulo de controle.

A interface do 'Sistema de Controle' é apresentada em um formulário amarelo claro. No topo, o título 'Sistema de Controle' é exibido em azul. Abaixo dele, a seção 'Localizar Veículo' contém o campo 'Informe a Placa do Veículo:' seguido de um campo de texto branco e um botão 'Pesquisar' cinza. Centralizado na interface é um grande retângulo branco vazio, provavelmente para exibir resultados. Na base da interface, há uma seção com campos de entrada rotulados: 'Placa:', 'Renavam:', 'Chassi:', 'Proprietario:', 'ID\_Veiculo:' e 'Tag ID:'. À direita desses campos, sob o título 'Opções', estão quatro botões: 'Excluir', 'Sair', 'Alterar' e 'Teste'.

Figura. 5.4 – Módulo de Controle

Fonte: Autor

### 5.3 - Resultados da aplicação do Modelo

Como resultado da aplicação do projeto foi desenvolvido o software que possui a interface demonstrada na figura 5.5, que integra o módulo de cadastro, o módulo de monitoramento e o módulo de controle em um ambiente só para o usuário.

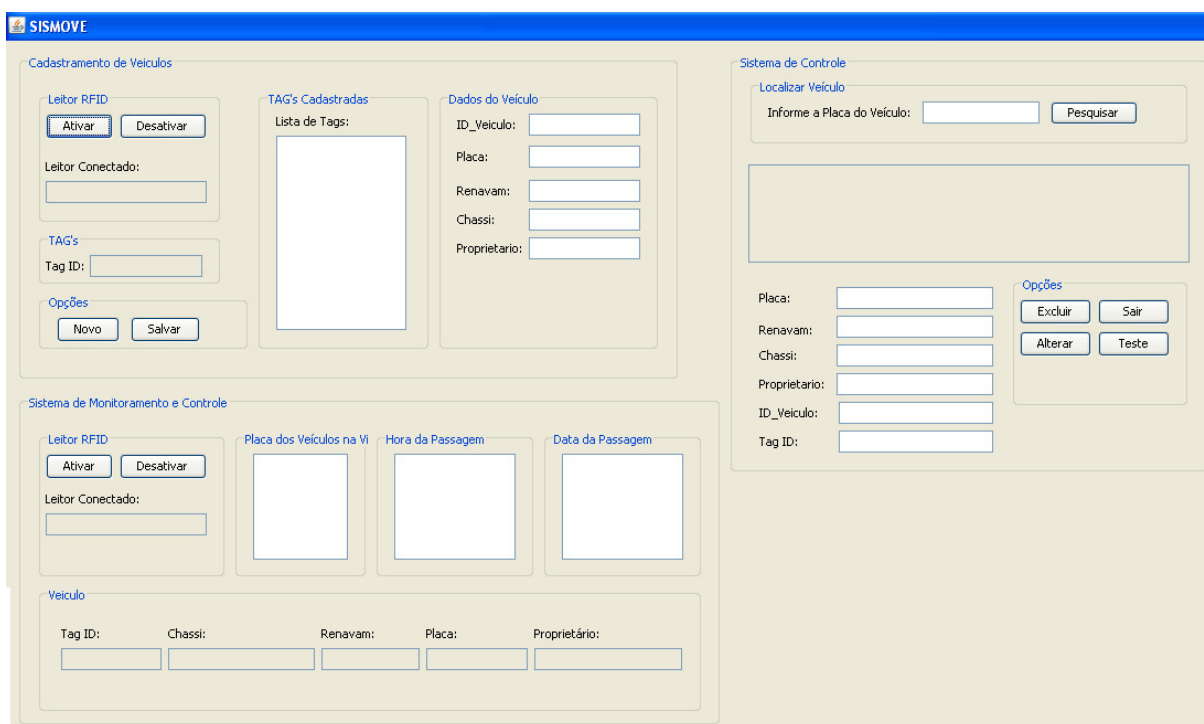


Figura 5.5 – Interface do *Software*

Fonte: Autor

O sistema de monitoramento de veículos permitirá controlar as funcionalidades propostas utilizando a maquete como modelo. As funcionalidades são: o cadastramento de veículos, o monitoramento da via e o controle dos veículos cadastrados com as opções de alteração e exclusão de um veículo.

A maquete que será utilizada como modelo é demonstrada na figura 5.6. Na imagem pode-se observar o leitor RFID e as miniaturas de veículos utilizadas cada qual com sua etiqueta RFID.

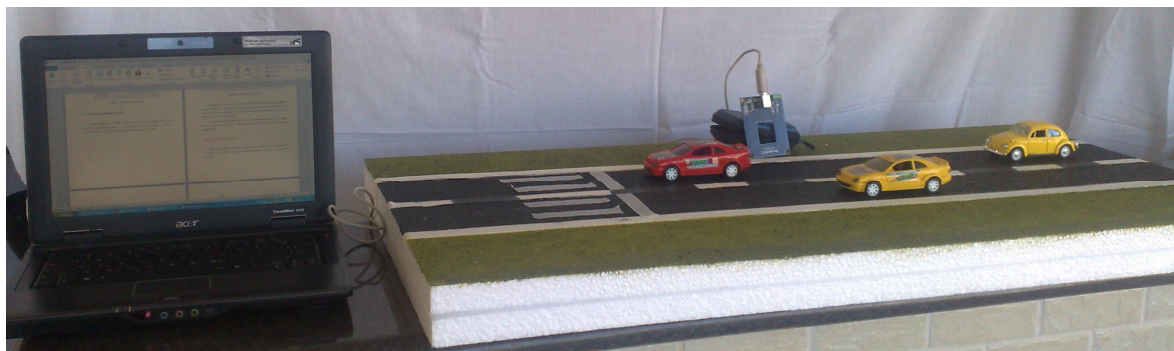


Figura 5.6 – Maquete Utilizada

Fonte: Autor

Para a identificação dos veículos foram anexados em cada miniatura etiquetas RFID compatíveis com o leitor. A figura 5.7 mostra a miniatura do veículo e a etiqueta RFID respectiva.



Figura 5.7 – Etiqueta RFID anexada ao veículo.

Fonte: Autor

Como o campo de leitura do leitor RFID é aproximadamente de 7 centímetros apenas, a etiqueta RFID deve ficar em local no veículo que não gere problemas para a sua correta identificação.

Como foi dito no capítulo 4 a conexão do leitor RFID e o computador onde o sistema desenvolvido está em execução se dá via porta USB, como mostra a figura 5.8.

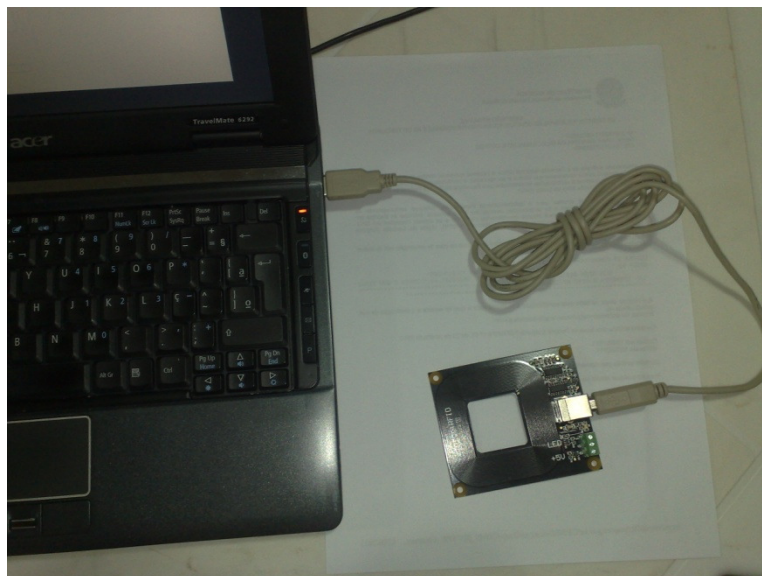


Figura 5.8 – Conexão do leitor RFID com o computador

Fonte: Autor

Para representar o projeto desenvolvido será primeiramente demonstrada a lista de veículos que já possuem cadastro no banco de dados, cada qual com sua etiqueta de identificação RFID e seus respectivos dados. Em seguida será simulado o cadastramento de um veículo com dados reais no banco de dados e anexado uma etiqueta RFID com um ID único que o acompanhará no restante da apresentação.

Após isto será demonstrado um exemplo de como é feita a identificação do veículo no sistema desenvolvido. Para tanto, será simulado, através da maquete, uma via com o trânsito de veículos. Dessa forma assim que o veículo passar diante do leitor RFID será identificado no banco de dados e seus dados disponibilizados na interface do *software*.

#### 5.4 - Custos do Projeto

Para o desenvolvimento do projeto e a apresentação de um protótipo, foi necessário escolher qual tecnologia seria utilizada na identificação dos veículos, a linguagem de

programação e a plataforma que seria desenvolvido o software e o sistema gerenciador de banco de dados que iria armazenar as informações dos veículos.

Como tecnologia utilizada para identificar cada veículo escolhida foi a radiofrequência, foi necessário adquirir um kit com leitor e etiquetas RFID. A escolha se baseou basicamente no custo e na facilidade de encontrar informações sobre o hardware. Sendo assim o kit RFID da marca Canadense Phidget foi escolhido.

O valor do kit é de USD 72,10 (setenta e dois dólares e dez centavos), contendo um leitor RFID e dez etiquetas RFID de vários formatos. A taxa cobrada para entrega é de USD 84,05 (oitenta e quatro dólares e cinco centavos), totalizando um valor de USD 156,15 (cento e cinquenta e seis dólares e quinze centavos) para adquirir o kit RFID da marca canadense Phidget (o valor em reais seria algo em torno de R\$ 256,08 - duzentos e cinquenta e seis reais e oito centavos - mais o imposto cobrado por compras no exterior).

Em contato com colegas que utilizaram a tecnologia RFID em projetos finais foi possível encontrar o leitor RFID escolhido. Dessa forma não foi necessário ter gastos para adquirir um leitor RFID.

Para desenvolver o software de controle utilizado foi utilizado a plataforma de desenvolvimento NetBeans em sua versão 6.9.1. Como a plataforma de desenvolvimento escolhida é de código aberto, ou no seu termo em inglês “open source”, foi necessário apenas entrar no site netbeans.org e baixar a última versão.

Da mesma forma foi escolhido o sistema gerenciador de banco de dados. Utilizando o MySQL Server 5.1.

## **5.5 - Avaliação Global do Modelo Proposto**

O modelo apresentado atende aos objetivos iniciais do projeto, que são identificar de forma única cada veículo que trafega na via utilizando a tecnologia RFID e disponibilizar os principais dados do veículo em um sistema desenvolvido para este fim.

Com o sistema de monitoramento sendo aplicado nas vias de trânsito de hoje e armazenando o histórico, como é proposto, é possível registrar por onde determinado veículo passou, bem como, a data e hora da passagem. Sendo o sistema utilizado em larga escala, ou

seja, em vários pontos de uma cidade, por exemplo, pode-se monitorar veículos que são roubados ou furtados.

O modelo desenvolvido tem sua aplicabilidade limitada a simulação da via modelada na maquete. Para um ambiente convencional seria necessário o desenvolvimento de um protótipo mais robusto e completo, no que diz respeito a programação do *software* e do banco de dados. O sistema abrange o monitoramento de uma via publica de trânsito e, assim, acaba permitindo que sendo utilizado em larga escala possa monitorar a entrada e saída de veículos de uma determinada cidade.

No projeto como um todo existem pontos fortes e pontos fracos. Os pontos fortes são o monitoramento da via utilizando RFID, a flexibilidade trazida por tecnologias moveis e a agilidade no trazida na identificação de veículos em movimento, sem a necessidade para-los. O ponto fraco mais importante é a limitação quanto a leitura de etiquetas pelo leitor RFID e a distancia de leitura. Uma vez que o kit utilizado possui essas características.



## **CAPÍTULO 6 – CONCLUSÃO**

### **6.1 - Conclusões**

A infraestrutura de transporte é um componente essencial para o desenvolvimento de um país, uma vez que possibilita o fluxo de bens e de pessoas, permitindo, entre outros suprir a demanda por mercadorias.

Uma infraestrutura inadequada eleva os custos logísticos, diminuindo a competitividade no mercado internacional e, dessa forma, tornando-se um entrave ao crescimento. Além disso, costuma gerar maiores tempos e custos das viagens, números de acidentes e níveis de emissão de poluentes.

Nesse contexto, uma infraestrutura de transporte de qualidade é fundamental para garantir uma eficiência logística capaz de permitir ao Brasil o desenvolvimento de todo seu potencial socioeconômico.

Com base nesse pressuposto e considerando que avaliação do cenário nacional aponta para um severo grau de carência do Brasil em termos de transportes, o projeto propõe uma solução que atue em uma pequena área da infraestrutura dos transportes: o monitoramento de vias de trânsito. Consequentemente, será possível apoiar o gerenciamento, planejamento e fiscalização do trânsito nas cidades Brasileiras.

Aliado a estes fatores a solução apresentada, consegue ainda, aumentar a segurança para os proprietários de veículos automotores, visto que o protótipo apresentado verifica os dados do veículo sem a necessidade de para-lo, dessa forma, aumentam-se as chances de localizar e reter o veículo em situações irregulares, ou até mesmo, em caso de roubo e furto.

Para alcançar este objetivo maior foram necessários determinar objetivos menores que deveriam ser alcançados antes. Com isso foi necessário o desenvolvimento do sistema de monitoramento que foi integrado à tecnologia RFID e funcionou como o planejado no início deste trabalho.

Os resultados obtidos, com os equipamentos que foram disponibilizados, como o kit RFID da Phidget, foram satisfatórios, uma vez que todos os veículos que foram previamente

cadastrados e seus dados armazenados, foram identificados com sucesso, pelo sistema desenvolvido para tal finalidade.

## **6.2 - Sugestões Para Trabalhos Futuros**

No desenrolar deste projeto muitas outras ideias foram sendo concebidas. O campo em que se encontra o projeto é muito abrangente. A tecnologia RFID esta sendo bastante estudada e aplicada em diversas áreas por trazer muitas vantagens em comparação a outras tecnologias já existentes.

Indo além do campo de apenas identificar os veículos em transito, as sugestões para trabalhos futuros estão citadas a seguir:

- Integrar o RFID com GPS (Sistema de Posicionamento Global). Uma das opções seria inicializar o monitoramento do veículo a partir do momento que este passa diante do leitor RFID;
- Implementar a característica de gestão de pagamentos dos impostos como, por exemplo, IPVA, gerando um alarme caso seja identificada alguma irregularidade; e
- Ser utilizado como controlador de velocidade em vias urbanas entre trajetos maiores, fazendo o uso de uma média de tempo gasto para percorrer a distancia entre dois leitores RFID.

## REFERÊNCIAS BIBLIOGRÁFICAS

DEITEL, HARVEY M. JAVA. **Como Programar**. 10ª Edição. Rio de Janeiro, 2010. Prentice Hall, - Br.

FINKENZELLER, KLAUS. **RFID Handbook, Fundamentals and Applications in Contactless Smart Cards and Identification**, 2ª Edição. New Jersey, Wiley 2009.

GLOVER, Bill; BHATT, Himanshu. **Fundamentos de RFID**. Rio de Janeiro, Atlas, 2007.

HUNT, V. DANIEL; PUGLIA, ALBERT; PUGLIA, MIKE. **RFID, A Guide to Radio Frequency Identification**. New Jersey, Wiley, 2007.

MANZANO, JOSE AUGUSTO N. G. **Mysql 5 - Interativo - Guia Básico de Orientação e Desenvolvimento**. São Paulo, 2009. Erica.

MEDICE, SILVIO. Elementos de Controle, Associação Brasileira de Monitoramento e Controle Eletrônico de Trânsito. 2010. Disponível em: [www.abramcet.com.br/elementos](http://www.abramcet.com.br/elementos). Acessado em: 03/03/2011.

MILANI, ANDRÉ. **MySQL – Guia do Programador**. São Paulo, 2009. Novatec.

MILES, STEPHEN B.; SARMA, SANJAY E.; WILLIAMS, JOHN R. **RFID Technology and Applications**. Cambridge, Cambridge University Press 2010.

PEIXOTO, CRESO DE FRANCO. Placa eletrônica é o sinal para pedágio urbano. 2010. Disponível em: <http://www.abramcet.com.br/Artigo.asp?ArtigoAtivo=126>. Acessado em: 03/03/2011.

PLANO DE LOGÍSTICA DA CONFEDERAÇÃO NACIONAL DE TRANSPORTES, 2010.

SAKAMOTO, JULIANA. **RFID – Radio Frequency Identification**. 2009. Disponível em: [http://www.gta.ufrrj.br/grad/09\\_1/versao-final/rfid/historico.html](http://www.gta.ufrrj.br/grad/09_1/versao-final/rfid/historico.html). Acessado em: 10/04/2011.

SIERRA, KATHY; BATES, BERT. **Use a Cabeça! Java**. 2ª Edição. Rio de Janeiro, 2007. O'Reilly.

SWEENEY, PATRICK J. **RFID for Dummies**. Indiana, Wiley 2008.

## APENDICE A – Conexão com Middleware RFID

```

public class TMiddlewareSM {

    private RFIDPhidget rfidSM;

    // Listeners RFID:
    private TAttachListenersSM attach_listener;
    private TDetachListenerSM detach_listener;
    private TTagGainListenerSM tagGain_listener;
    private TTagLossListenerSM tagLoss_listener;
    private TErrorListenerSM error_listener;

    private String appError;

    public void startRFID( JTextField jtfAtachadoSM, JTextField jtfTagSM, JTextField
jtfPlacaSM, JTextField jtfRenavamSM, JTextField jtfChassiSM, JTextField
jtfProprietarioSM, JTextArea jtaListaSM) {
        // public void startRFID(JCheckBox jcbAntenaSM, JTextField jtfAtachadoSM,
JTextField jtfTagSM, JTextArea jtaListaSM) {
            try {

                rfidSM = new RFIDPhidget();

                attach_listener = new TAttachListenersSM( jtfAtachadoSM);
                detach_listener = new TDetachListenerSM( jtfAtachadoSM);
                tagGain_listener = new TTagGainListenerSM(jtfTagSM,jtfPlacaSM,jtfRenavamSM,
jtfChassiSM,jtfProprietarioSM,jtaListaSM);
                tagLoss_listener = new TTagLossListenerSM(jtfTagSM);
                error_listener = new TErrorListenerSM(this.appError);

                rfidSM.addAttachListener(attach_listener);
                rfidSM.addDetachListener(detach_listener);
            }
        }
    }
}

```

```

        rfidSM.addErrorListener(error_listener);
        rfidSM.addTagGainListener(tagGain_listener);
        rfidSM.addTagLossListener(tagLoss_listener);

        rfidSM.openAny();

    } catch (PhidgetException ex) {

    }
}

public void stopRFID( JtextField jtfAtachadoSM) {

    jtfAtachadoSM.setText("");
    try {

        rfidSM.removeTagLossListener(tagLoss_listener);
        rfidSM.removeTagGainListener(tagGain_listener);
        rfidSM.removeErrorListener(error_listener);
        rfidSM.removeDetachListener(detach_listener);
        rfidSM.removeAttachListener(attach_listener);

        rfidSM.close();

        rfidSM = null;
    }
}

```

## APENDICE B – Função “Conectar” do RFID

```

public class TAttachListenersSM implements AttachListener{

    JTextField jtfAtachadoSM;

    public TAttachListenersSM(JTextField jtfAtachadoSM){

        this.jtfAtachadoSM = jtfAtachadoSM;
    }

    public void attached(AttachEvent ae){
        try {
            RFIDPhidget attached = (RFIDPhidget)ae.getSource();

            jtfAtachadoSM.setText(attached.getDeviceName()+Boolean.toString(attached.isAttached()));

            attached.setAntennaOn(true);

        } catch (PhidgetException e) {
            System.out.println(e.getDescription()+"Phidget error "+e.getErrorNumber());
        }
    }
}

```

## APENDICE C – Função “Leitura da Etiqueta RFID” do RFID

```

public class TTagGainListenerSM implements TagGainListener{

    private JTextField jtfTagSM,jtfPlacaSM,jtfRenavamSM, jtfChassiSM, jtfProprietarioSM;
    private JTextArea jtaListaSM;

    List<String> oldTagID = new ArrayList<String>();

    public TTagGainListenerSM(JTextField jtfTagSM,JTextField jtfPlacaSM, JTextField
jtfRenavamSM, JTextField jtfChassiSM, JTextField jtfProprietarioSM, JTextArea
jtaListaSM){
        this.jtfTagSM = jtfTagSM;
        this.jtfPlacaSM = jtfPlacaSM;
        this.jtfRenavamSM = jtfRenavamSM;
        this.jtfChassiSM = jtfChassiSM;
        this.jtfProprietarioSM = jtfProprietarioSM;
        this.jtaListaSM = jtaListaSM;

    }

    public void tagGained(TagGainEvent tagGainEvent){
        jtfTagSM.setText(tagGainEvent.getValue());
        preencheCampos(tagGainEvent.getValue());

        System.out.println(tagGainEvent.getValue());
    }

    private void preencheCampos(String tag) {

        SQL SQLSM = new SQL();
        Veiculo veiculo = new Veiculo();

```



```

String tagID = tag;
// if(SQLSM.consulta(tagID)){
//   jtfPlacaSM.setText(" ");

    veiculo = SQLSM.localizar(tagID);

    jtfPlacaSM.setText(veiculo.getPlaca());

    jtfRenavamSM.setText(veiculo.getRenavam());
    jtfChassiSM.setText(veiculo.getChassi());
    jtfProprietarioSM.setText(veiculo.getProprietario_id());

    jtaListaSM.append(veiculo.getPlaca() + "\n");

    oldTagID.add(tagID);

//   }else{
//       JOptionPane.showMessageDialog(null, "Veiculo não
cadastrado", "ERRO!", JOptionPane.ERROR_MESSAGE);
//   }

    }catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Erro SQL!", "Erro SQL!",
JOptionPane.ERROR_MESSAGE);
    }

}
}

```

**APENDICE D – Função “Desconectar” do RFID**

```
public class TDetachListenerSM implements DetachListener{

    JTextField jtfAtachadoSM;

    public TDetachListenerSM( JTextField jtfAtachadoSM){

        this.jtfAtachadoSM = jtfAtachadoSM;
    }

    public void detached(DetachEvent de){
        jtfAtachadoSM.setText("");
    }
}
```

**APENDICE E – Função “Error” do RFID**

```
public class TErrorListenerSM implements ErrorListener{

    private String appError;

    /** Creates a new instance of RFIDErrorListener */
    public TErrorListenerSM(String appError){
        this.appError = appError;
    }

    public void error(ErrorEvent errorEvent){
        System.out.println(appError + errorEvent.toString() + "RFID Error Event");
    }

}
```

## APENDICE F – Módulo de Cadastro, Monitoramento e Controle.

```

public class SQL {

    // todas as rotinas para acessar, modificar, alterar o banco de dados

    private Connection conexao; // cria uma variavel (conexao) do tipo connection.

    public SQL() throws SQLException {
        this.conexao = (Connection) ConexaoBD.getConexao(); // cria uma conexão com o
        banco de dados
        //Connection conexao = ConexaoBD.getConexao();
    }

    // Metodos de Rotina de Banco de Dados:

    // ----- Método Adicionar no BD -----

    public void adiciona(Veiculo v1) throws SQLException {
        String sql = "insert into veiculo (veiculo_id, placa, renavam, chassi, proprietario_id)"
            + "values (?, ?, ?, ?, ?)";
        // declaração de uma string de nome sql com o valor entre aspas dupla.

        PreparedStatement stmt = (PreparedStatement) conexao.prepareStatement(sql);
        //conexao pois é necessário conectar ao BD.
        // faz com que a string sql
        // seja executada no banco de dados. Cria um objeto do tipo preparedStatement de nome
        // stmt. O código é enviado para o mysql e executado lá.

        // stmt.setString(1, p1.getProprietario_cpf());
        stmt.setString(1, v1.getVeiculo_id());
        stmt.setString(2, v1.getPlaca());
        stmt.setString(3, v1.getRenavam());
        stmt.setString(4, v1.getChassi());
        stmt.setString(5, v1.getProprietario_id());

        stmt.execute();
        stmt.close();

    }

    // ----- Lista todos os dados da Tabela Veiculos -----

```

```

public List <Veiculo> getLista(String veiculo_id) throws SQLException {
    String sql = "select * from veiculo where veiculo_id like ?";
    PreparedStatement stmt = (PreparedStatement) this.conexao.prepareStatement(sql);

    stmt.setString(1,veiculo_id);

    ResultSet rs = stmt.executeQuery(); // rs recebe todos os contatos da query stmt.

    List<Veiculo> ListaVeiculo = new ArrayList<Veiculo>();

    while (rs.next()) { // result set next para ir enquanto existe o proximo no BD.

        Veiculo v1 = new Veiculo();
        v1.setVeiculo_id(rs.getString("veiculo_id"));
        v1.setPlaca(rs.getString("placa"));
        v1.setRenavam(rs.getString("renavam"));
        v1.setChassi(rs.getString("chassi"));
        v1.setProprietario_id(rs.getString("Proprietario_id"));
        v1.setCodTag(rs.getString("cod_tag"));

        ListaVeiculo.add(v1);

    }
    rs.close();
    stmt.close();
    return ListaVeiculo;
}

// -----

// ----- Alterar um dado no BD -----

public void altera(Veiculo veic254) throws SQLException{

    String sql = "update veiculo set cod_tag=?, placa=?, renavam=?, chassi=?,
    proprietario_id=? where veiculo_id=?";

    PreparedStatement stmt = (PreparedStatement) conexao.prepareStatement(sql);

    stmt.setString(1, veic254.getCodTag());
    stmt.setString(2, veic254.getVeiculo_id());
    stmt.setString(3, veic254.getPlaca());
    stmt.setString(4, veic254.getRenavam());
    stmt.setString(5, veic254.getChassi());
    stmt.setString(6, veic254.getProprietario_id());

```

```

        stmt.execute();
        JOptionPane.showMessageDialog(null, "Dados do Veiculo alterado com Sucesso!");

    }

    public List<Veiculo> getLista() {
        throw new UnsupportedOperationException("Not yet implemented");
    }

    // -----
    /* public List<Veiculo> getLista1(String nome) throws SQLException {
        System.out.println("AQUI");
        String sql = "select * from veiculo where IDTag like ?";
        PreparedStatement stmt = (PreparedStatement) this.conexao.prepareStatement(sql);
        stmt.setString(1, nome); // filtra a pesquisa com a informação que estiver na string nome
        // ouseja, substitui ? pela informação da string.
        //Recebo como parametro a string nome.
        ResultSet rs = stmt.executeQuery();

        List<Veiculo> minhaLista = new ArrayList<Veiculo>();

        while (rs.next()) {

            Veiculo v1 = new Veiculo();
            v1.setPlaca(rs.getString("Placa"));
            v1.setChassi(rs.getString("Chassi"));
            v1.setRenavam(rs.getString("Renavam"));
            v1.setCategoria(rs.getString("Categoria"));
            v1.setMunicipio(rs.getString("Municipio"));
            v1.setUf(rs.getString("UF"));
            v1.setSituacao(rs.getString("Situacao"));
            v1.setRestricao(rs.getString("Restricao"));
            v1.setNacionalidade(rs.getString("Nacionalidade"));
            v1.setIDTag(rs.getString("IDTag"));

            minhaLista.add(v1);

        }
        rs.close();
        stmt.close();
        return minhaLista;
    }

    */
    // ----- Cadastrar TAG no BD -----

```

```

public void cadastrar(Veiculo veiculo) throws SQLException{
    //PreparedStatement para inserir

    String sql = "insert into veiculo ( veiculo_id, placa, renavam, chassi,
    proprietario_id, cod_tag) values (?,?,,?,?,?)";
    PreparedStatement stmt = (PreparedStatement) conexao.prepareStatement(sql);
    //Seta valores
    stmt.setString(1, veiculo.getVeiculo_id());
    stmt.setString(2, veiculo.getPlaca());
    stmt.setString(3, veiculo.getRenavam());
    stmt.setString(4, veiculo.getChassi());
    stmt.setString(5, veiculo.getProprietario_id());
    stmt.setString(6, veiculo.getCodTag());

    //Executa e fecha
    stmt.execute();
    stmt.close();
}

public Boolean consulta(String tagID) throws SQLException {

    String sql = "select placa from veiculo where cod_tag = ?";
    PreparedStatement stmt = (PreparedStatement) conexao.prepareStatement(sql);

    stmt.setString(1, tagID);
    stmt.execute();

    ResultSet rs = stmt.executeQuery();

    if(rs.next()){
        rs.close();
        stmt.close();
        return true;
    }else{
        rs.close();
        stmt.close();
        return false;
    }

}

public Veiculo localizar(String tagID) throws SQLException{

    String sql = "select * from veiculo where cod_tag = ?";
    PreparedStatement stmt = (PreparedStatement) conexao.prepareStatement(sql);

    stmt.setString(1, tagID);
    ResultSet rs = stmt.executeQuery();

```

```

        rs.next();

        Veiculo v1 = new Veiculo();
        v1.setVeiculo_id(rs.getString("veiculo_id"));
        v1.setPlaca(rs.getString("placa"));
        v1.setRenavam(rs.getString("renavam"));
        v1.setChassi(rs.getString("chassi"));
        v1.setProprietario_id(rs.getString("Proprietario_id"));

        rs.close();
        stmt.close();

        return v1;
    }

    public void remover(Veiculo vex) throws SQLException {

        String sql = "delete from veiculo where veiculo_id=?";
        PreparedStatement stmt = (PreparedStatement) conexao.prepareStatement(sql);
        stmt.setString(1, vex.getVeiculo_id());
        stmt.execute();
        stmt.close();

    }

    public void alterar(Veiculo valt) throws SQLException {

        String sql = "update veiculo set cod_tag=?, placa=?, renavam=?, chassi=?,
        proprietario_id=? where veiculo_id=?";

        PreparedStatement stmt = (PreparedStatement) conexao.prepareStatement(sql);

        stmt.setString(1, valt.getCodTag());
        stmt.setString(2, valt.getVeiculo_id());
        stmt.setString(3, valt.getPlaca());
        stmt.setString(4, valt.getRenavam());
        stmt.setString(5, valt.getChassi());
        stmt.setString(6, valt.getProprietario_id());

        stmt.execute();
        JOptionPane.showMessageDialog(null, "Dados do Veiculo alterado com Sucesso!");
    }

}

```



## APENDICE G – Conexão com o Banco de Dados

public class ConexaoBD { // classe criaconexão que vai servir como link do banco de dados e o Java

```

    // Metodo Estatico que passa um objeto do tipo connection
    public static Connection getConexao() throws SQLException {
        try {
            Class.forName("com.mysql.jdbc.Driver"); // Seta para o drive do MySQL
            System.out.println("Conectado ao Banco"); // Mensagem informadno a conexcção ao
BD.
            return DriverManager.getConnection("jdbc:mysql://localhost/sismove", "root",
"thnepo"); // Conecta ao BD.
        } catch (ClassNotFoundException e) {
            throw new SQLException(e.getMessage());
        }
    }
}

```